

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Génération automatique de descriptions de structure de base de données dans un système de documentation

Weygers, René

Award date:
1979

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX, NAMUR

INSTITUT D'INFORMATIQUE

Année Académique 1978-1979

**GÉNÉRATION AUTOMATIQUE
DE DESCRIPTIONS DE STRUCTURE
DE BASE DE DONNÉES
DANS UN SYSTÈME
DE DOCUMENTATION**

Mémoire présenté en vue de
l'obtention du grade de
Licencié et Maître en Informatique

RENÉ WEYGERS

FACULTES
UNIVERSITAIRES
N.-D. DE LA PAIX
NAMUR

Bibliothèque

FM B 16

1979/17

FM B 16 / 1979 / 17

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX, NAMUR

INSTITUT D'INFORMATIQUE

Année Académique 1978-1979

**GÉNÉRATION AUTOMATIQUE
DE DESCRIPTIONS DE STRUCTURE
DE BASE DE DONNÉES
DANS UN SYSTÈME
DE DOCUMENTATION**

Mémoire présenté en vue de
l'obtention du grade de
Licencié et Maître en Informatique

RENÉ WEYGERS

BS 321 2936



GS20-27022

"Tout ce que je sais,
c'est que je ne sais rien"

SOCRATE

AVANT PROPOS

Je remercie les personnes qui, à divers titres, contribuèrent à la réalisation de ce mémoire et plus particulièrement :

Mr CLARINVAL, cicerone averti dont le précieux parrainage fut vivement apprécié lors de la rédaction du mémoire.

Mrs OLIVIER et GERADIN, guides de tous les instants lors de l'élaboration de la proposition technique.

Mr CRUCIFIX, responsable du centre informatique où le stage fut effectué.

L'ensemble du corps professoral de l'institut d'enseignement supérieur de Namur et de l'institut d'informatique qui ont parfait ma formation humanitaire par une ouverture éclairée sur la réalité informatique.

Table des Matières

AVANT - PROPOS

TABLE DES MATIERES

INTRODUCTION

CHAPITRE 1 : LE SYSTEME DE DOCUMENTATION

1.1. Définition et objectif d'un système de documentation	1
1.2. Système de documentation du CTISM	1
1.2.1. Contenu du système de documentation du CTISM	2
1.2.2. Organisation du " " " " "	4
1.2.3. Fonction " " " " "	6
1.2.4. Fonctionnement du " " " "	7

CHAPITRE 2 : LE PROBLEME ETUDIE

2.1. Phases de résolution	8
2.2. Intérêt de la génération automatique	10

CHAPITRE 3 : LE LANGAGE DE DESCRIPTION ORGANIQUE DE BASE DE DE DONNEES

3.1. Rappel	12
3.2. Syntaxe du langage	13
3.3. Equivalence des niveaux de structure	14
3.4.1. Clauses descriptives	17
3.4.2. Placement des phrases	18
3.4.3. Remarques	20
3.4.4. Travail de l'analyseur spécifique	21

CHAPITRE 4 : LES TABLES DE RANGEMENT DU LANGAGE DE DESCRIPTION ORGANIQUE DE BASE DE DONNEES

4.1. Structure des tables	24
4.2. Enchaînement logique des tables	27
4.3. Remarque sur les tables	29

CHAPITRE 5 : LE GENERATEUR DE DESCRIPTIONS DE STRUCTURE DE BASE DE DONNES UDS/V2

5.1. Entrée du générateur	30
5.2. Logique " "	30
5.3. Composition du générateur	31
5.4. Module d'édition PRGENSKE	31
5.4.1. Entrées du module PRGENSKE	31
5.4.1.1. Squelettes descriptifs de la structure de base de données UDS/V2	31
5.4.1.2. Tables du générateur	34
5.4.2. Sortie du module PRGENSKE	35
5.4.3. Travail du module PRGENSKE	36
5.5. Ecriture des squelettes	36
5.6. Opérationnalité du générateur	39

CHAPITRE 6 : ETUDE COMPARATIVE AVEC LES LOGICIELS SCAPFACE ET LEXICON

6.1. Le logiciel SCAPFACE	40
6.2. Le logiciel LEXICON	43
6.3. Améliorations techniques	47
CONCLUSION	49
REFERENCES	51
ANNEXES	53

Introduction

Ce mémoire présente une proposition de réalisation de la génération automatique de descriptions de structure de **base** de données dans le cadre du système de documentation du "Centre de traitement l'information du service médical (CTISM)" de l'armée belge.

Le premier chapitre est consacré à la définition du système de documentation tandis que le second aborde la définition du problème traité par ce mémoire. Les trois chapitres suivants présentent l'ensemble des apports nécessaires pour la réalisation de la génération. Ces apports sont : un langage de description organique de base de données, un format de tables où ranger les informations du langage, un générateur de descriptions de structure de base de données.

Le dernier chapitre tente une étude comparative entre la solution présentée et celles des logiciels SCAPFACE et LEXICON.

Il convient de préciser que le système de documentation du CTISM est à l'état de projet et que ce mémoire ne constitue qu'une proposition de solution pour un automate de génération.

Chapitre 1 : Le système de documentation.

1.1 Définition et objectif d'un système de documentation.(5,6,8,9)

Un système de documentation est la réunion d'un répertoire centralisé de descriptions de données (utilisables à tout moment par tout projet) et d'un éventail de programmes d'enregistrement, d'interrogation, de contrôle et de génération qui gèrent ce répertoire.

Tout système de documentation repose sur un modèle ou sur une terminologie convenue que manipule un langage spécialisé.

Le concept de système de documentation pallie une lacune mainte fois constatée : l'insuffisance de communication de la documentation qui entraînait des redondances et des inconsistances et dont résultaient une maintenance pénible et une intégrité aléatoire des données.

Un système de documentation permet à tout responsable de constater l'état du système d'information, d'en réduire le temps et le coût de maintenance, de développer des projets informatiques et de les suivre à chaque étape de leur cycle de vie.

L'objectif essentiel d'un système de documentation est d'ordre économique : accroître la qualité du travail de tout centre informatique (sa productivité).

1.2. Système de documentation du CTISM.

Le système de documentation du CTISM est inspiré d'ISDOS (11,12). Grâce à un appareillage diversifié, il assiste les phases de conception et d'analyse fonctionnelle, qui sont les étapes initiales du cycle de vie de tout projet informatique, mais aussi les étapes suivantes d'analyse organique, de test, d'exploitation et de maintenance. Il s'est fixé deux objectifs primordiaux : l'automatisation de la documentation et la génération de programmes.

1.2.1. Contenu du système de documentation du CTISM.

A la base de tout système de documentation existe un langage qui permet à l'utilisateur de définir son problème, tant sous l'angle des traitements que sous l'angle des données.

Le système de documentation du CTISM utilise le PSL (problem statement language) d'ISDOS dans une version remaniée. Il distingue dans tout système d'information un ensemble d' OBJETS (êtres) qui possèdent entre eux des relations.

Objets du système de documentation du CTISM. (11,12)

Les objets sont répartis en classes.

1°) objets_de_l'environnement

INPUT collection d'informations produites à l'extérieur du système mais utilisés par celui-ci.

OUTPUT collection d'informations produites par le système mais utilisées à l'extérieur de celui-ci.

INTERFACE objet nommé qui interagit entre le système et l'extérieur.

2°) objets_descriptifs_des_aspects_statiques_du_problème

SET collection d'occurrences d'objets porteurs de valeur :
"entity, output, input"

PROCESS toute fonction, tout traitement réalisés dans le système

ENTITY collection d'informations produites et utilisées à l'intérieur du système

GROUP collection d'"élément(s)" et/ou "groupe(s)"

ELEMENT la plus petite unité d'information (indivisible)

RELATION connexion logique nommée entre deux "entités"

3°) objets_descriptifs_des_aspects_dynamiques_du_problème.

EVENT description des occurrences possibles lors du fonctionnement du système

CONDITION description d'un aspect de l'état du système

INTERVAL description d'une unité de temps.

4°) objets_de "management"

exemple :

PROBLEM-DEFINER personne qui définit le problème

5°) propriétés_des_objets

exemple :

ATTRIBUTE caractéristiques particulières d'un objet (qui ne peuvent être indiquées par d'autres clauses du PSL)

Relations dans le système de documentation du CTISM (11,12).
.....

Les relations répondent à des approches différentes (mais complémentaires) de tout système d'information

Les relations sont réparties en classes.

1°) flux_du_système : interaction entre le système et son environnement

exemple : PROCESS P GENERATES OUTPUT-1
 " P RECEIVES INPUT -1

2°) structure_du_système : hiérarchie des traitements

exemple : PROCESS SUBPARTS ARE P1 P2 P3

remarque : la relation réciproque est générée automatiquement pour toute relation

$$\text{PROCESS} \left\{ \begin{array}{c} p1 \\ p2 \\ p3 \end{array} \right\} \text{ PART OF } p$$

3°) structure des données : hiérarchie des données utilisées et/ou manipulées par le système

exemple : ENTITY E CONSISTS OF GROUP 1 ELEM1 ELEM2

4°) dérivation des données : manipulation ou dérivation des données par le système

exemple : PROCESS DERIVES SET-1

5°) taille du système : facteurs influençant le volume de traitements nécessaires

exemple : SET S CARDINALITY IS 100

6°) aspect dynamique : relations du système dans le temps

exemple : PROCESS P INCEPTION-CAUSES EVENT-1

7°) aspect propriété : propriétés de chaque objet

exemple : ENTITY E DESCRIPTION ...

8°) gestion de projet : documentation sur les personnes qui réalisent et utilisent le système

exemple : ENTITY E RESPONSIBLE-PROBLEM-DEFINER IS
THE BOSS

1.2.2. Organisation du système de documentation du CTISM.

Support de mémorisation.

Le support de mémorisation est une base de données UDS/V2 (SIE-MENS).

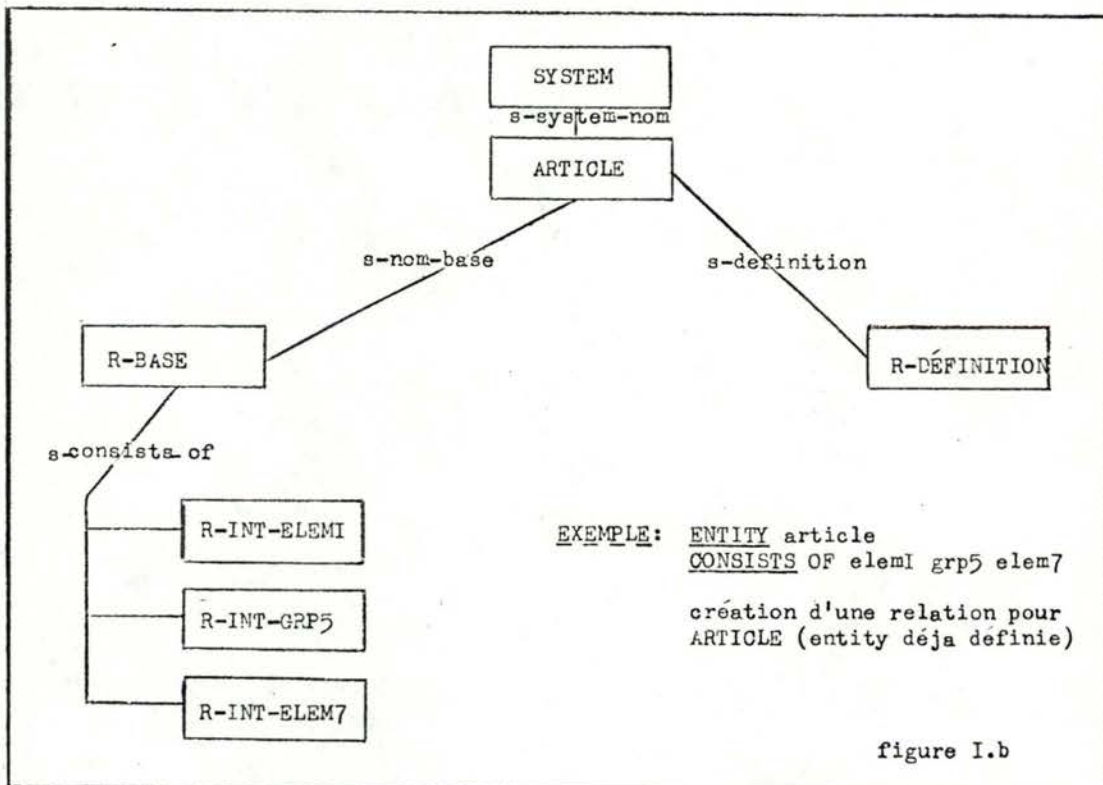
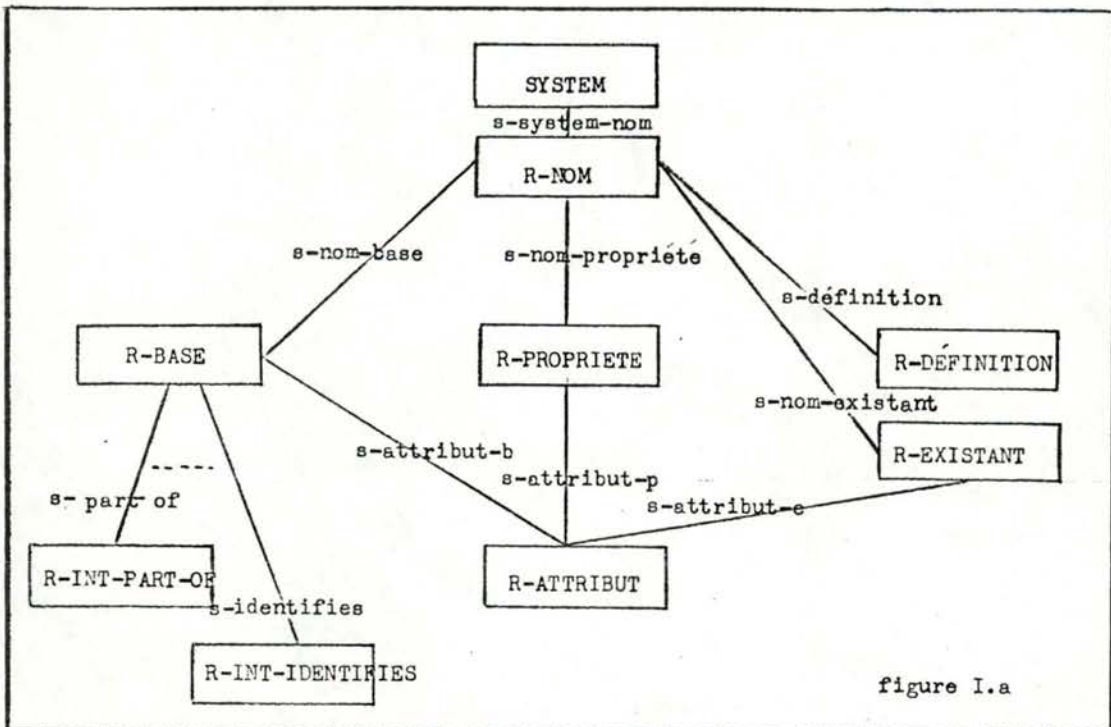
Un outil tel qu'un SGBD (système de gestion de base de données) offre l'avantage de posséder un langage de manipulation répondant aux impératifs de la gestion d'un système de documentation (8,9) : création, modification, suppression, consultation. La forme de cette base de données (fig. 1.1) fait de tout OBJET un ensemble de "RECORD": BASE (longueur, type,...), PROPRIETES, NOM, ATTRIBUT, DEFINITION, EXISTANT (si l'objet existait avant la mise en place du système de documentation) et de toute relation un SET (au sens de CODASYL, voir chapitre 3). Il est inutile de pousser plus avant l'étude de cette base de données car elle ne possède aucun impact direct sur le sujet du mémoire.

Description d'un objet (et de ses relations).

.....

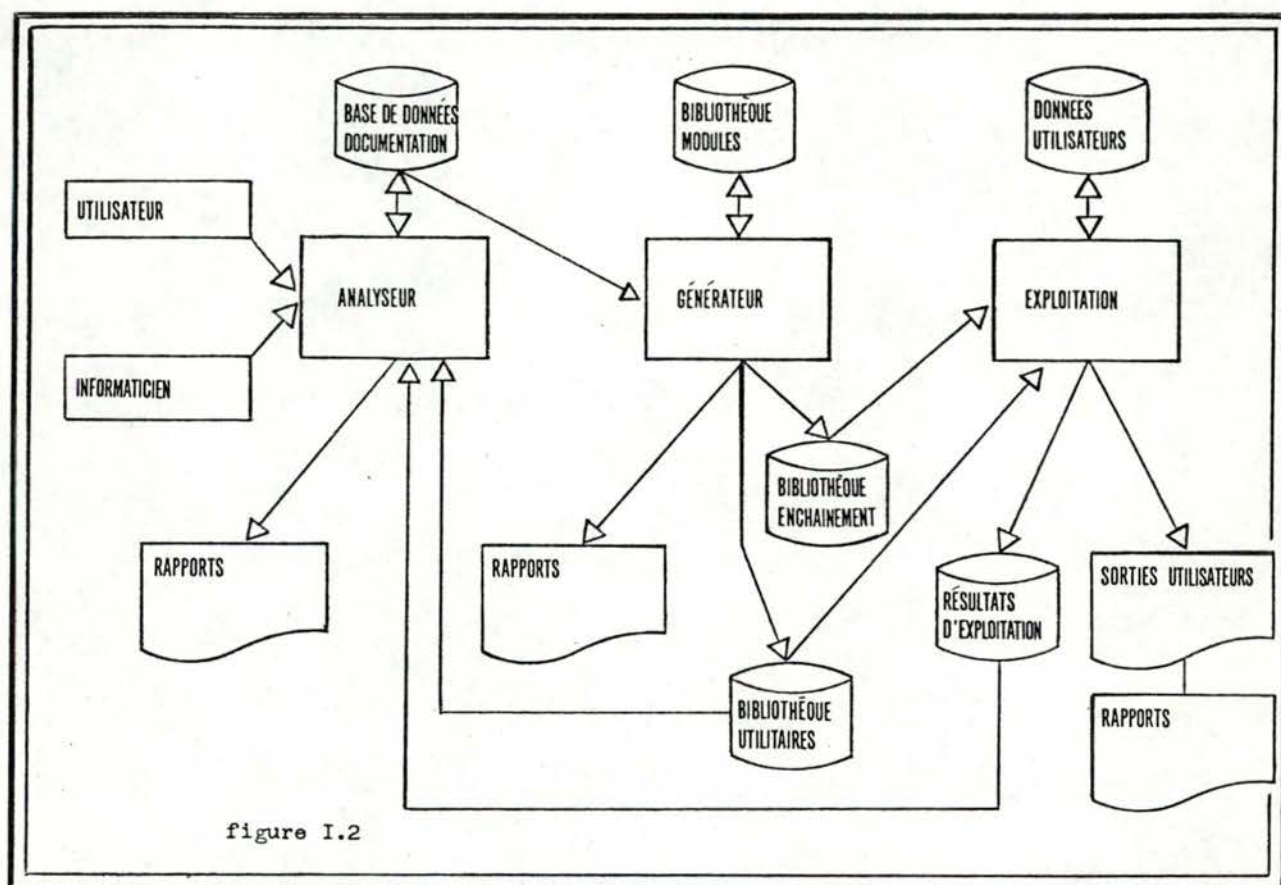
Toute description d'objet est rédigée en PSL sous forme de sections composées de clauses où seules une identification et une définition de l'objet sont obligatoires.

Toute clause peut être créée, ajoutée, modifiée à l'aide d'un langage de commande analogue à celui d'ISDOS.



1.2.3. Fonction du système de documentation du CTISM.

La fonction du système de documentation du CTISM est schématisée par la figure 1.2.



Les fonctions de l'ANALYSEUR sont

1. l'analyse syntaxique du texte rédigé en PSL
2. l'analyse sémantique et l'aide à la mise au point
3. l'accès à la base de données DOCUMENTATION

Les fonctions du GENERATEUR sont :

1. la création de programmes et leur stockage en librairie
2. la mise à jour des programmes et librairies
3. l'optimisation des procédures

1.2.4. Fonctionnement du système de documentation du CTISM.

L'utilisateur décrit son problème en langage PSL ou effectue des modifications.

L'analyseur, grâce à des tables de définition du langage PSL, vérifie la syntaxe et la sémantique des phrases en PSL. Il les codifie sous une forme simplifiée et effectue le traitement adéquat sur la base de données DOCUMENTATION : chargement, extraction, modification.

Le générateur grâce à des descriptifs de sortie (squelettes) et aux informations qu'il puise dans la base de données DOCUMENTATION produit les sorties désirées à savoir les programmes (tant la partie des déclarations de données que la partie des instructions) qu'il stocke en librairie.

Un documenteur permet (à tout instant) à l'utilisateur d'obtenir tous les rapports qu'il désire sur l'état du système de documentation.

Chapitre 2 : Le problème étudié.

Tout programme comporte ou fait référence à une description des données qu'il utilise.

Dans un système de documentation, toute description des données doit être présente dans la base de données DOCUMENTATION avant d'être employée. Aussi, peut-on permettre à un outil programmé (GENERATEUR) de puiser cette description et de la traduire dans le langage de programmation voulu.

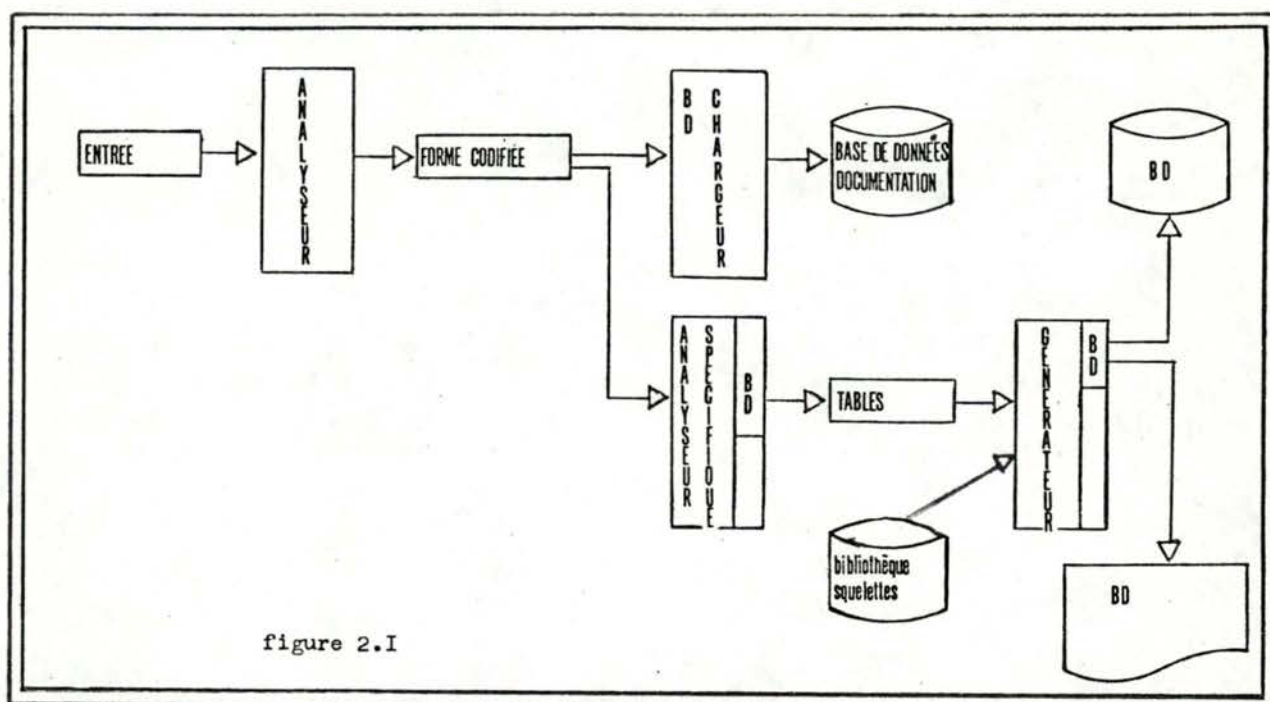
Le problème qui nous retiendra est la génération automatique de descriptions de structure de base de données.

2.1. Phases de résolution.

La génération automatique de descriptions de structure de base de données nécessite l'addition de cinq compléments au système de documentation du CTISM :

1. Un langage de description organique de base de données, vu comme une extension du PSL existant. Ce langage est fait de clauses des SGBD envisagés et de renseignements permettant d'une part de réaliser une correspondance entre les niveaux de structure et, d'autre part, de retrouver, grâce au seul PSL, le "schéma" et le(s) "sous-schéma(s)" à générer. (voir 3.1)

2. Un analyseur spécifique (aux bases de données) dont le rôle est mis en évidence à la figure 2.1.



Les phrases du langage PSL sont traitées par l'analyseur général qui les codifie. Le chargeur de la base de données range cette forme codifiée dans la base de données DOCUMENTATION (à l'aide du schéma de la base de données DOCUMENTATION et des instructions de manipulation).

Les phrases codifiées sont aussi transmises à un analyseur spécifique qui les décompose en unités logiques d'information qu'il range dans des tables à destination de générateurs spécifiques (aux bases de données).

L'analyseur spécifique effectue la reconnaissance des phrases codifiées grâce à des tables de travail où il retrouve les diverses séquences autorisées. Ces tables contiennent les mots réservés du langage et leurs suites possibles.

L'analyseur spécifique est réalisé par un responsable du système de documentation du CTISM et n'est pas l'objet d'un chapitre de ce mémoire.

3. Des tables de rangement où l'analyseur spécifique peut ranger les unités logiques d'informations à destination des générateurs spécifiques. Ces tables permettent de disposer, à tout moment, en un seul lieu et sous une forme simple de toutes les informations concernant une base de données et évitent aux générateurs de devoir accéder à la base de données DOCUMENTATION. Leur format respecte les niveaux de structuration des données du système de documentation du CTISM à savoir : SET, ENTITY, GROUP, ELEMENT, RELATION.

4. Un générateur spécifique de descriptions de structure de base de données UDS/V2. Il retrouve le modèle de cette description dans une bibliothèque de squelettes et puise l'information variable propre à une base de données particulière dans les tables évoquées au point précédent.

5. Des squelettes de description de structure de base de données UDS/V2. Ces squelettes sont des descriptifs de la forme sous laquelle doit être décrite toute base de données UDS/V2.

2.2. Intérêt de la génération automatique.

Sans vouloir dresser une liste exhaustive des avantages de la génération automatique, il me semble intéressant d'en énumérer quelques-uns.

Le système de documentation oblige l'utilisateur à introduire toutes ces informations en une seule entrée, ce qui permet à l'administrateur de données (responsable de la cohérence et de l'intégrité du système de documentation) d'effectuer son travail avec le maximum de sécurité. Cette remarque est valable aussi bien pour la création que la modification d'une description de base de données.

La description d'une base de données est unique dans tout le système d'information. Sans le mécanisme de génération automatique, l'utilisateur doit non seulement la décrire en PSL mais aussi dans le langage du SGBD choisi, ce qui peut entraîner des discordances de description.

La description d'une base de données est fiable car elle est vérifiée syntaxiquement et sémantiquement par l'analyseur spécifique et le générateur spécifique. Ces vérifications sont effectuées lors de l'introduction de la description de la base de données et permettent à l'utilisateur de corriger et réintroduire cette dernière avant de la compiler.

La modification d'une description de base de données bénéficie des avantages évoqués ci-dessus mais, de plus, permet la détection des impacts directs de son intervention.

L'analyse et la génération automatique sont réalisées sans intervention humaine ce qui accroît la sécurité du travail.

A l'avenir, un langage de description de traitements devrait permettre la génération automatique des instructions d'un programme dans un langage évolué (exemple : PROCEDURE DIVISION de COBOL). Le système de documentation possède la description des données d'un programme. Aussi, paraît-il aberrant de ne pas réaliser dès à présent la génération de la description des données d'un programme dans un langage évolué (exemple : DATA DIVISION de COBOL).

Chapitre 3 : Le langage de description organique de base de données.

Pour couvrir le concept de base de données, les responsables du système de documentation du CTISM se sont refusés à créer de nouveaux objets ou de nouvelles relations. Les ajustements nécessaires ont été réalisés par l'incorporation au PSL de clauses spécifiques.

Ce langage de description organique de base de données est constitué de phrases que l'utilisateur place dans la description de l'objet idoine (au niveau des "SYSTEM PROPERTIES").

Ces phrases sont conçues dans une volonté de lisibilité et de facilité de prise en charge par un outil programmé.

Le système de documentation du CTISM doit disposer d'un matériel de la firme SIEMENS. Toutes les clauses du SGBD UDS/V2 (voir A5.1, A5.2) de cette firme sont couvertes par le langage. De plus, ce SGBD étant de la famille CODASYL, toutes les clauses prévues par la norme CODASYL sont elles aussi couvertes par le langage.

3.1. Rappel

Il importe de déterminer l'exacte signification des termes utilisés par la norme CODASYL(1).

SCHEMA description complète d'une base de données, nommant et décrivant toutes les "areas", les "sets", les "records" et les "data-items" et/ou "data-aggregates" qui leur sont associés

SUB-SCHEMA description partielle d'une base de données ne reprenant (et parfois modifiant) du schéma que ce qui est utile à un (ou plusieurs) programme spécifique

AREA subdivision nommée (de l'espace mémoire adressable dans la base de données) contenant les occurrences de "records" et "sets"

RECORD collection nommée de 0 à n "data-aggregates" et/ou "data-items" (il y a de 0 à n occurrences de "record" dans une base de données)

DATA-AGREGATE collection nommée de "data-items" se présentant sous deux formes possibles : vecteur (unidimensionnel) ou groupe répétitif (constitué de "data-item(s)" et/ou groupe(s) répétitif(s))

DATA-ITEM plus petite unité d'information nommée dont l'occurrence est une valeur.

SET collection nommée de "records" ; chaque occurrence de "set" lie une occurrence d'un "record" (dit "OWNER") à une ou plusieurs occurrences de "record" (dit "MEMBER").

3.2. Syntaxe du langage.

Le langage étant destiné à être traité par un analyseur doit respecter une règle précise.

Afin de permettre l'identification de la phrase traitée, toute phrase doit débiter par un mot réservé connu de l'analyseur. Ce dernier possède une table de mots réservés et de leurs suites possibles, ce qui l'autorise à vérifier la cohérence syntaxique mais aussi sémantique de toute phrase.

Le langage est composé de phrases, elles-mêmes constituées d'une suite finie de mots. Ces mots sont de trois types.

1. Mot réservé : mot obligatoire qui permet à l'analyseur d'identifier la phrase mais aussi son contenu

exemple : LOCATION, RELATION

2. Mot facultatif : sans valeur d'information, ce mot est placé à des fins de lisibilité de la phrase

exemple : LOCATION MODE IS

3. Mot représentant une catégorie d'opérandes (qui aura une valeur bien précise à l'écriture de la phrase)

exemple : RELATION IS relation-name

AMBIGUITE

Tout mot de type 1 est suivi d'un ou plusieurs mots de type 3. Malheureusement, certaines options (= parties de phrase) sont facultatives et il devient impossible de déterminer la ou les option(s) prise(s) parmi l'éventail disponible.

exemple : RESERVED (IN set-name)(IS set-name)....

Les mots IN et IS sont ignorés par l'analyseur car ils sont de type 2. Il faut placer un mot réservé dans chaque option facultative devant chaque mot de type 3 facultatif afin de lever l'ambiguïté.

exemple : RESERVED (IN SET set-name)(IS set-name)(ON entity-name)..
INCORRECT car le problème est déplacé à l'option suivante

RESERVED (IN SET set-name)(IS ONLY set-name)....

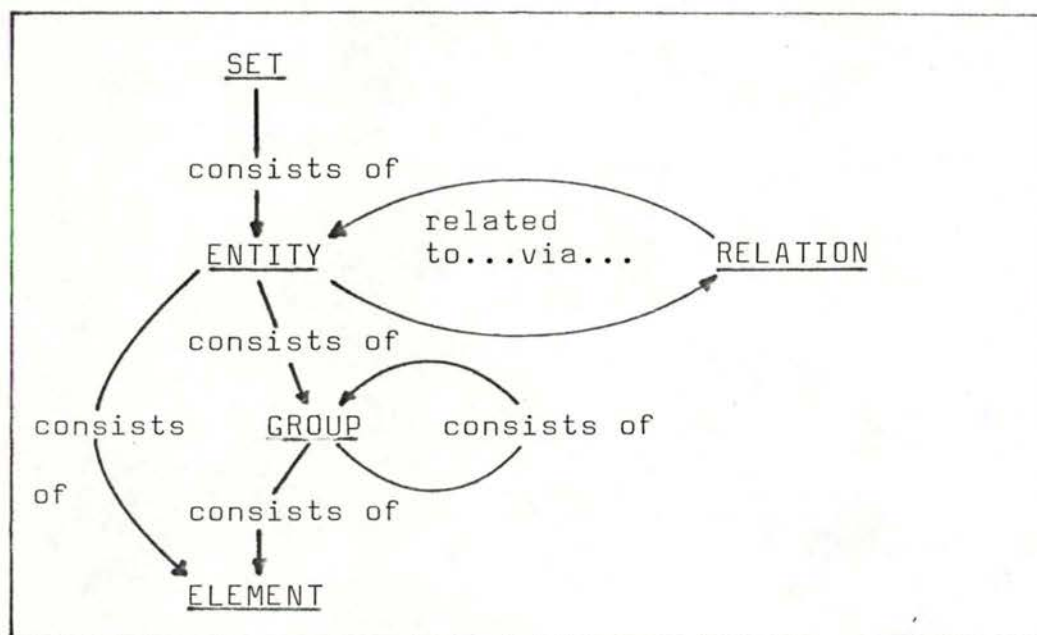
CORRECT

3.3. Equivalence des niveaux de structure.

Les niveaux de structure de données des SGBD envisagés doivent trouver leur équivalent dans la structure de données du système de documentation afin de pouvoir placer les phrases

(du langage) aux niveaux correspondants.

La structuration CODASYL (et donc UDS/V2) a été précisée ci-avant. La structuration des données du système de documentation est la suivante.



Le "SET" est une notion fonctionnelle qui doit être affinée pour répondre aux exigences organiques de la norme CODASYL.

On distingue alors trois types de "SET" : "SCHEMA, SUB-SCHEMA, AREA".

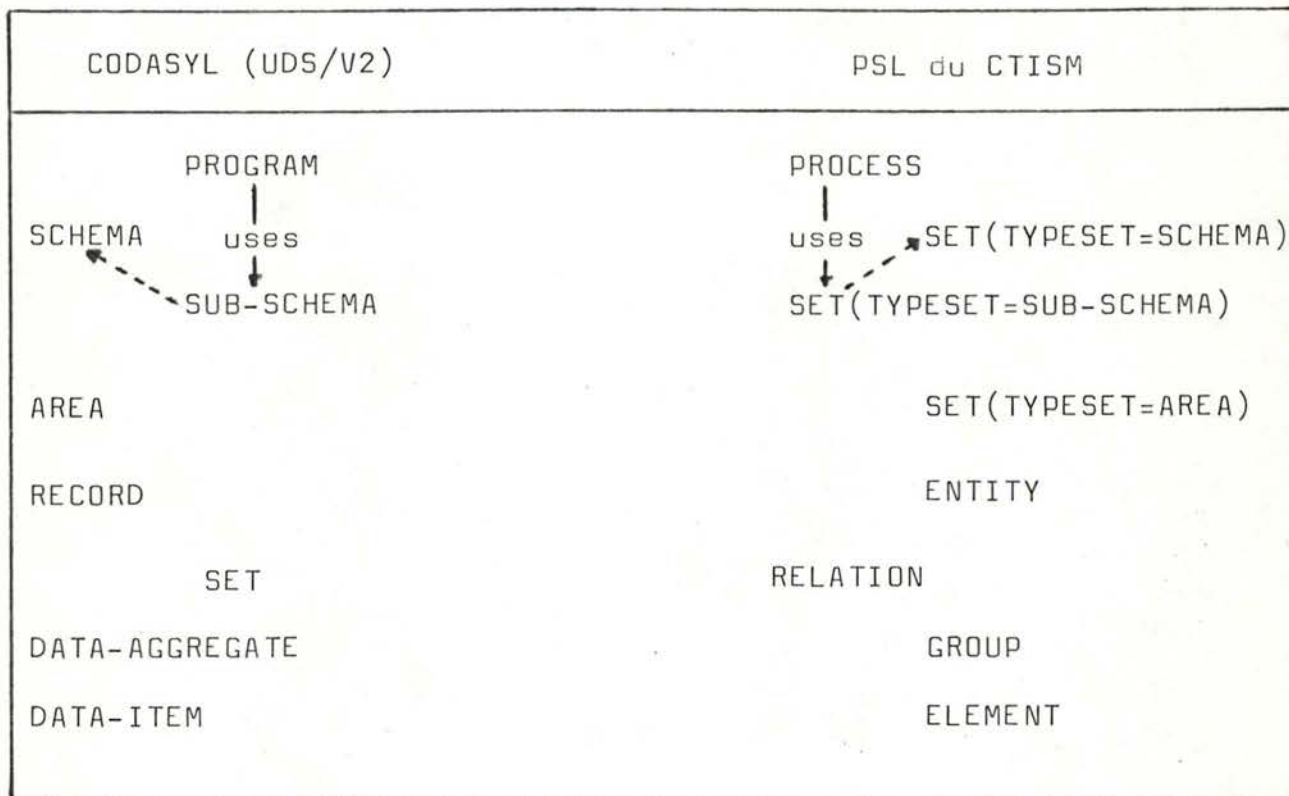
Cette distinction se fait par l'adjonction au langage de la phrase "TYPESET IS

$$\left\{ \begin{array}{l} \underline{\text{SCHEMA}} \\ \underline{\text{SUB-SCHEMA}} \\ \underline{\text{AREA}} \\ \underline{\text{OTHER}} \end{array} \right\} "$$

La "RELATION" au sens du système de documentation du CTISM est toujours implémentée sous forme de "SET" au sens de la norme CODASYL.

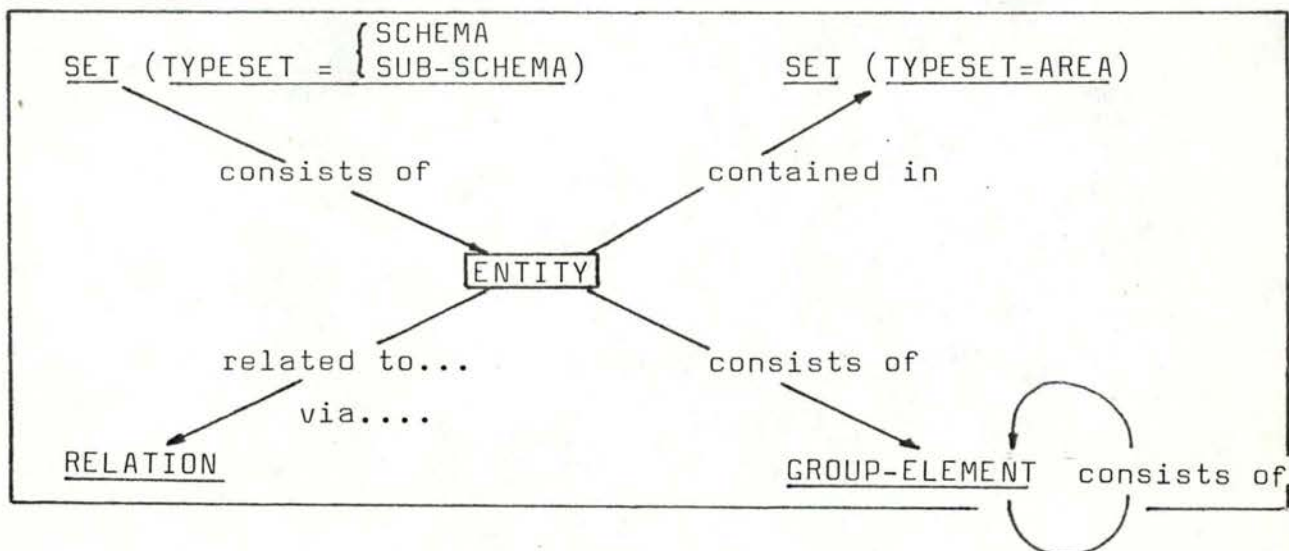
Le schéma ci-dessus est un squelette de modèle de description organique commun aux divers implémenteurs (UDS, ...)

L'équivalence des niveaux de structure peut être représentée comme suit.



Le lien $\cdots \rightarrow$ est défini par la phrase "SUB-SCHEMA OF set-name" que l'on écrit dans chaque relation de "SET (TYPESET=SUB-SCHEMA)" afin de préciser le "SET (TYPESET = SCHEMA)" auquel il se rapporte.

Les liens entre tous les autres niveaux sont repris au paragraphe 3.4.4. et ci-dessous est représentée la structure de description à utiliser.



3.4. Contenu du langage.

La structuration des données est définie par les phrases traditionnelles du PSL auxquelles se greffent d'autres phrases telles : "TYPESET", "SUBSCHEMA" évoquées précédemment.

3.4.1. Clauses descriptives.

Schéma
.....

Les clauses du schéma de la norme CODASYL et du SGBD UDS/V2 sont prises intégralement et traduites dans le langage sous une forme syntaxique adéquate. Il y a fusion des clauses provenant des deux normes.

exemple :

1. clause LOCATION MODE	DIRECT ...
	CALC ...
+ pour CODASYL	VIA ...
2. clause ORDER IS	FIRST LAST NEXT PRIOR
+ pour UDS/V2	IMMATERIAL
3. clause OCCURS ...	
clause PICTURE ...	
identiques pour UDS/V2 et CODASYL	
4. clause PLACEMENT ...	uniquement pour UDS/V2
clause RESULT...	uniquement pour CODASYL

Des différences apparaissent au niveau des restrictions d'écriture ou d'emploi d'une partie ou de la totalité d'une clause dans la norme CODASYL ou dans le SGBD UDS/V2 ; la présentation la plus restrictive a été adoptée.

exemple : MODE IS ... facultatif pour CODASYL
obligatoire pour UDS/V2

Sous-schéma

Certaines modifications ou ajouts valables uniquement pour tout sous-schéma viennent compléter les clauses descriptives (valables pour tout schéma) reprises ci-dessus.

exemple : ajout RENAMING ...
 PRIVACY_KEY ...

modification OCCURS TIMES DEPENDING_ON name

3.4.2. Placement des phrases.

L'équivalence de structure a été définie. On place les phrases relatives aux "SCHEMA, SUB-SCHEMA, AREA" au niveau du SET, celles du "RECORD" au niveau de l'ENTITY et celles du "SET (CODASYL)" au niveau de la RELATION.

Cependant, certaines clauses descriptives de certains objets sont placées dans la description d'autres objets. Il faut alors renseigner dans la clause descriptive l'objet réel qu'elle décrit.

Groupes et éléments

Les groupes et éléments peuvent appartenir à diverses entités et présenter des particularités de représentation organique selon l'entité à laquelle ils participent. Si l'on place toutes les phrases, relatives à ces particularités, au niveau du groupe et/ou élément, on doit mentionner pour chacune d'elles l'entité maîtresse en cause. On se trouve alors devant un (volumineux) ensemble hétéroclite difficile à traiter.

La solution à ce problème est de placer chaque phrase, relative

aux particularités, dans chaque entité maîtresse pour laquelle elle est valable.

Si une entité maîtresse ne présente aucune particularité dans ces groupes et éléments, elle utilise leurs propriétés habituelles.

L'analyseur spécifique peut en traitant l'entité ne pas ranger dans la base de données DOCUMENTATION la phrase lue car il sait qu'elle ne concerne pas cette entité mais un de ces groupes ou éléments.

Cette forme d'adressage est réalisée par l'adjonction à la phrase figurant dans la description de l'entité de " $\left\{ \begin{array}{l} \text{ON} \\ \text{FOR} \end{array} \right\} \left\{ \begin{array}{l} \text{group} \\ \text{element} \end{array} \right\} - \text{name} "$.

exemple : ENTITY E CHECK ON ELEM-1 IS PICTURE

Groupes et éléments dans un sous-schéma
.....

Les groupes et éléments peuvent appartenir à un schéma et/ou à un ou plusieurs sous-schémas. Aussi est-il indispensable de placer dans chacune des phrases (décrivant un groupe ou un élément) qui peuvent être utilisées pour le schéma et le sous-schéma une clause facultative de localisation sous la forme "IN SET sub-schéma-name".

exemple : ENTITY E CHECK ON ELEM-1 IN SET SUBSCHEMA-1 IS PICTURE

Entités dans un sous-schéma
.....

Cette distinction entre groupes et éléments d'un sous-schéma ou d'un schéma existe également au niveau de l'entité.

En effet, un sous-schéma reprend une partie des entités du schéma et dans ce cas, deux situations sont possibles.

1. L'entité appartenant au sous-schéma est identique à celle du schéma. Cela signifie qu'elle reprend la description complète

de l'entité pour le schéma et qu'elle est composée de l'intégralité des groupes et éléments définis dans la description de l'entité pour le schéma. Dans ce cas, les phrases (pour les groupes et éléments de cette entité) qui ne sont valables que pour un sous-schéma précis doivent posséder l'option "IN SET subschéma-name" et doivent être placées dans la rubrique définissant l'entité.

Une autre méthode consiste à généraliser l'emploi de la phrase évoquée au point suivant en créant une nouvelle entité pour chaque entité du schéma reprise dans un sous-schéma.

2. L'entité appartenant à un sous-schéma ne contient qu'une partie des groupes et éléments de cette même entité dans le schéma. Dans ce cas, l'utilisateur doit définir et nommer une nouvelle entité où il place les phrases relatives aux groupes et éléments de cette entité pour ce sous-schéma.

Néanmoins, le compilateur après l'analyse du schéma ne connaît au maximum que ce qu'il y a dans ce schéma. Aussi, s'il rencontre dans tout sous-schéma une entité nouvelle, elle est inconnue de lui et donc fautive.

Il faut préciser dans la définition de cette nouvelle entité le nom de l'entité du schéma dont elle est la représentation dans le sous-schéma. Ceci est réalisé en plaçant dans la description de la nouvelle entité la phrase "USERNAME IN SCHEMA IS entity-name".

Il incombe à l'analyseur spécifique de remplacer le nom de l'entité pour le sous-schéma par celui de l'entité du schéma.

3.4.3. Remarques.

1. Lorsqu'une entité d'un schéma est présente dans un sous-schéma, on ne désire pas toujours dans ce sous-schéma reprendre toutes les relations qui lient cette entité ni tous les "SET (TYPESET=AREA)" auxquels appartient cette entité.

On utilise alors la phrase "RESTRICTION FOR subschéma-name
CONCERN

$$\left\{ \begin{array}{l} \text{AREA set-name (, set-name)...} \\ \text{RELATION relation-name (, relation-name)...} \end{array} \right\} "$$

Cette phrase est placée dans la déclaration de l'entité et permet à l'analyseur spécifique de n'envisager pour le sous-schéma précisé que les relations et/ou "sets (area)" non mentionnés dans la phrase mais apparaissant dans la clause "RELATED TO ... VIA..." et/ou "CONTAINED IN ...".

2. Un numéro de version permet de distinguer la version d'un schéma ou d'un sous-schéma .

On peut ainsi avoir diverses versions pas forcément écrites dans le même langage de SGBD.

Quoi qu'il en soit ce numéro doit figurer au moins une fois dans chaque phrase où apparaît un nom d'objet.

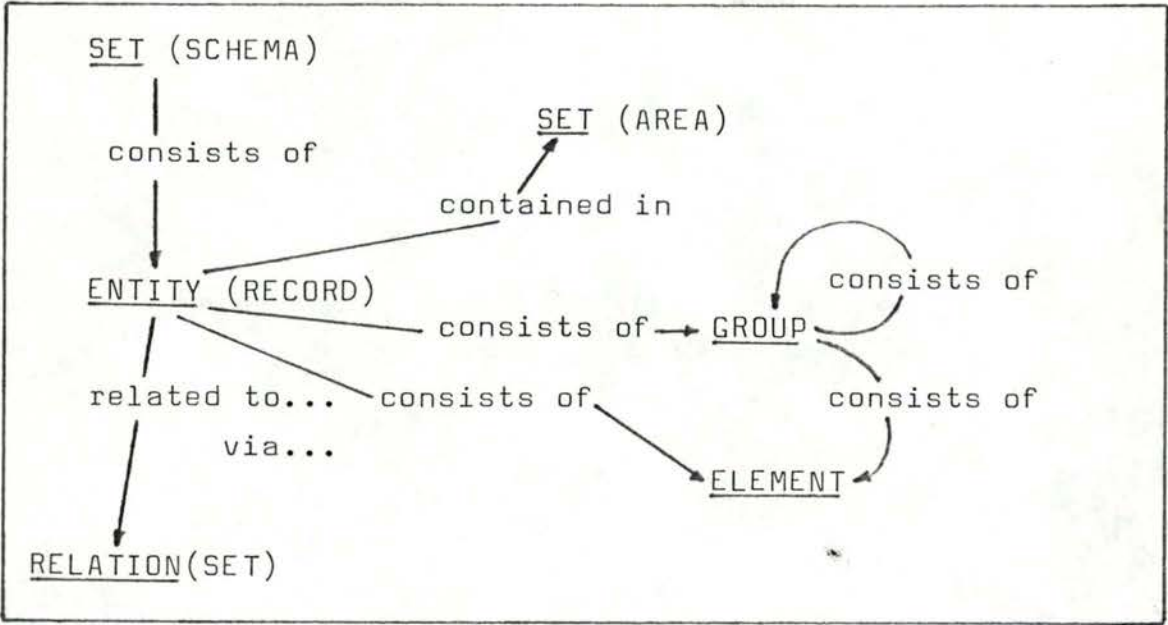
exemple : PICTURE FOR element-name OF version-number...

3.4.4. Travail de l'analyseur spécifique.

Recherche du "SET (SCHEMA)" et de ses composantes.
.....

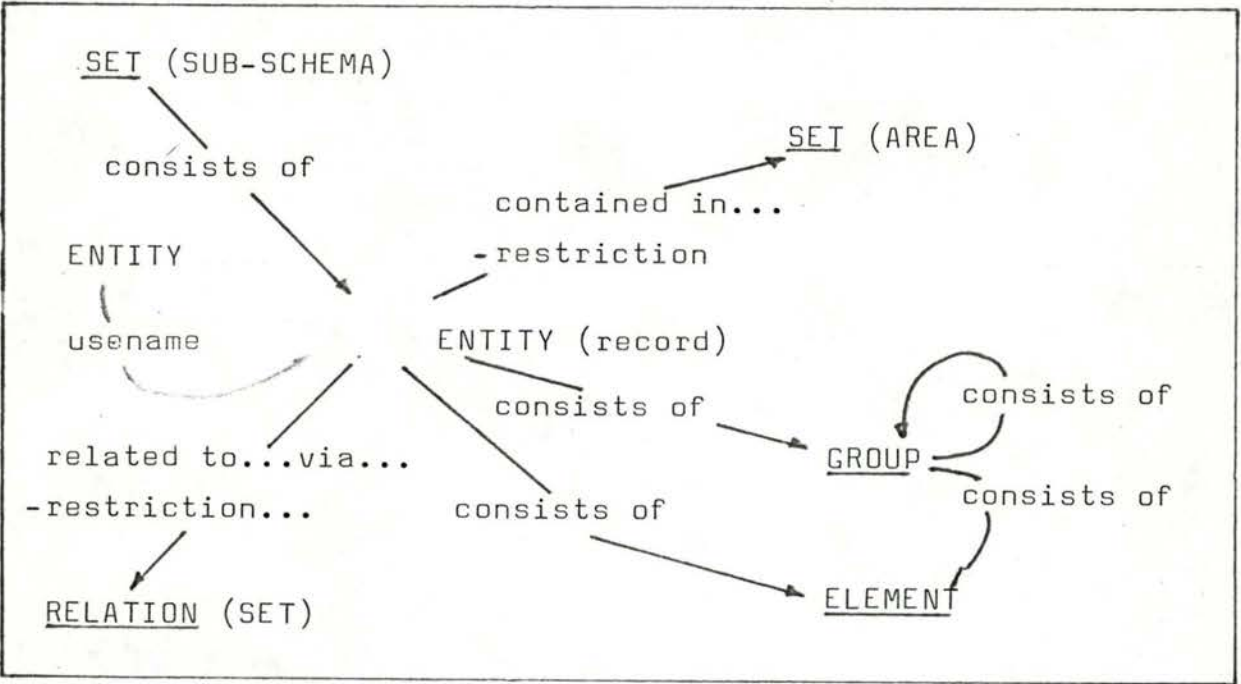
A l'aide des relations mentionnées, l'analyseur spécifique est apte à retrouver toute la structure du schéma.

Il suffit de lui préciser le point de départ, qui initialisera le parcours, à savoir le nom du schéma.



Recherche d'un "SET (SUB-SCHEMA)" et de ses composantes.
.....

A l'instar du schéma, il suffit de préciser le point de départ
qui est le nom du sous-schéma.



Décomposition des phrases en attributs.

Chaque phrase est traitée par l'analyseur ; elle est faite d'uni-

tés logiques d'information appelées "ATTRIBUTS".

La composition des phrases est précisée dans le descriptif du langage repris en A 1.1. à A.1.23.

exemple : A1.12 PICTURE FOR element-name OF version number

attribut obligatoire

IN SET set-name

attribut facultatif

IS string

attribut obligatoire

Ces attributs sont numérotés et respectent des règles précisées dans le descriptif du langage.

L'analyseur range leur valeur dans des tables à destination du générateur de descriptions de structure de base de données.

Le lecteur est invité à prendre brièvement connaissance du langage (A 1.1 à A 1.23).

Remarques : les numéros d'attributs du langage de description organique de base de données débutent au chiffre 300 car il existe dans le système de documentation environ 290 attributs préalablement définis pour d'autres utilisations.

Chapitre 4 : Les tables de rangement du langage de description organique de base de données.

L'analyseur spécifique traite les phrases du langage. Il les décompose en attributs qu'il représente dans des tables. Il ajoute des informations déjà présentes dans le PSL. La liste de ces informations déduites et transformées en attributs est reprise en page A.1.24.

exemple : liste du "CONTAINED IN area-name" devient une liste d'attributs numéro 424.

4.1. Structure des tables.

Forme et contenu

Les tables respectent la terminologie et la structuration des données du système de documentation, ce qui permet d'envisager la prise en compte d'autres SGBD sans devoir remanier la structure des tables.

La description de tout objet concernant une base de données est faite de phrases qui sont composées d'attributs. On a donc pour chaque type d'objet un ensemble maximal d'attributs possibles.

Une table d'index par type d'objet reprend le nom de l'objet ainsi que la liste des attributs pour ce type d'objet.

Une table d'information par type d'objet reprend les informations (valeurs) relatives aux attributs de ce type d'objet.

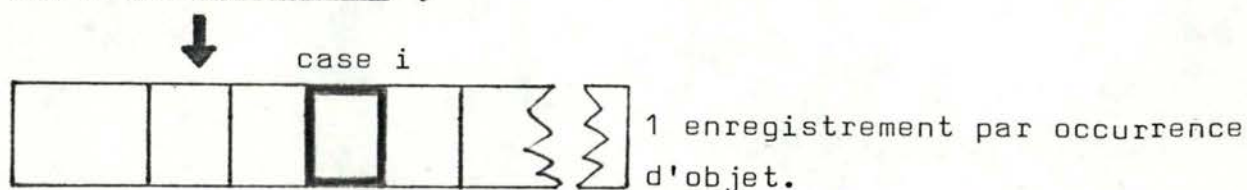
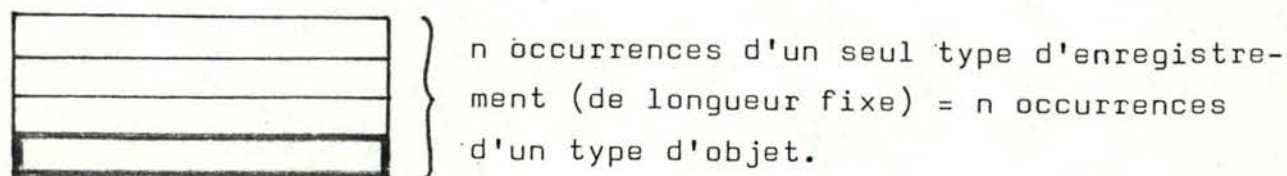
Ces tables sont constituées d'enregistrements de longueur fixe.

Tables d'information

}

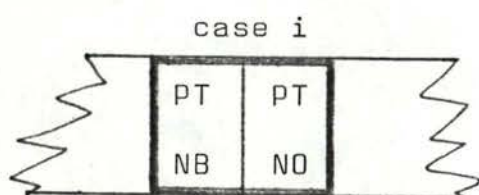
n occurrences d'un seul type d'enregistrement de longueur fixe.

Tables d'index



1 case par numéro d'attribut
(voir figure 4.1)

Chacune de ces cases est décomposée comme suit.

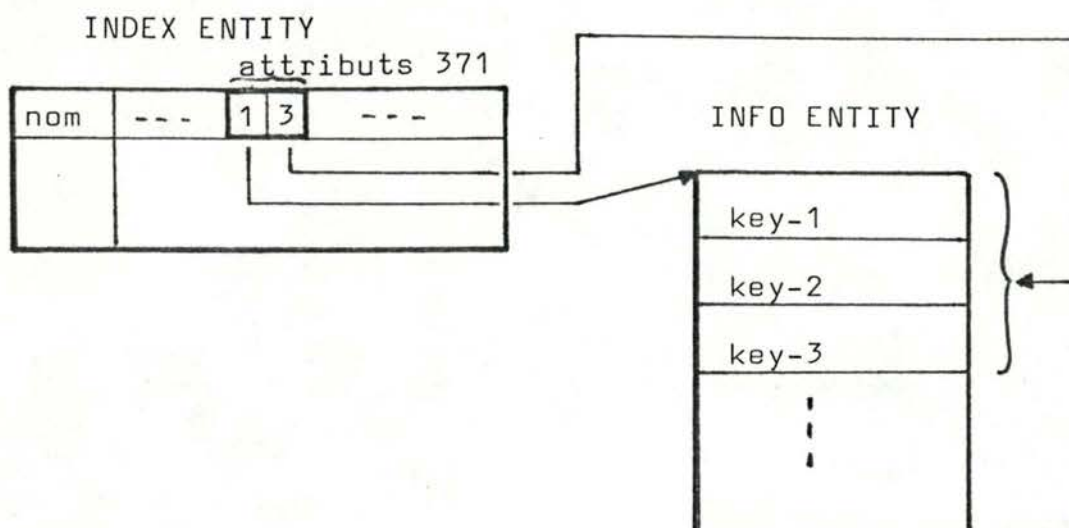


nombre d'occurrences de valeurs présentes pour l'attribut de la case i

pointeur vers la valeur de l'attribut correspondant à la case i (à 0 si pas garni), ce pointeur est l'adresse de début de la liste de(s) valeur(s) de l'attribut dans la table d'information.

exemple : SEARCH KEY IS key-1 key-2 key-3 ...

clé de SEARCH = attribut 371



Le mécanisme ci-dessus prend en charge les répétitions de valeurs d'attributs au sein d'une phrase.

Mais, il arrive que la phrase elle-même soit répétitive.

Une table d'index secondaire supprime ce problème de répétition.

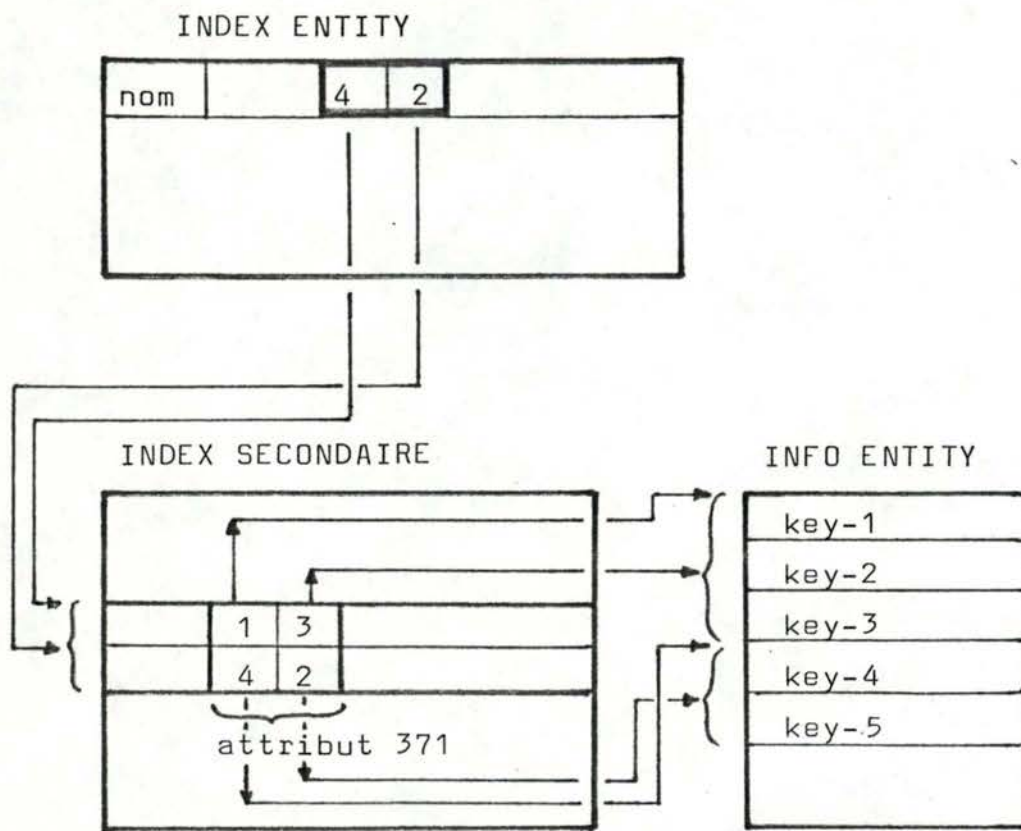
Table d'index secondaire

.....

Dans la table d'index de l'objet où une phrase est répétitive, les attributs de cette phrase sont remplacés par deux pointeurs. Le premier précise le nombre de répétitions de la phrase tandis que le second donne l'adresse de début dans l'index secondaire de(s) liste(s) d'attributs de cette phrase.

exemple : SEARCH KEY IS key-1 key-2 key-3...

SEARCH KEY IS key-4 key-5



Cet index secondaire est utilisé pour les phrases répétitives DBPRIVACY, CONDITION, INDEX, OPERATION, RENAMING, SEARCH. La liste des emplacements des attributs dans les index des différents types d'objets se trouve en figure 4.1.

4.2. Enchaînement logique des tables.

Une trame logique, maintenue grâce à des pointeurs, permet de retrouver dans les tables le schéma et le(s) sous-schéma(s) à générer.

4.2.1. Remarques générales.

A la lumière de la figure 4.2., on constate que les différentes formes de "SET (schéma, sub-schéma, area)" ont leurs renseignements placés dans une table INDEX SET et une table INFO SET. En effet, seuls quelques renseignements varient : au contraire

Set		Entity	Group-element	Relation	[Member]		SEARCH	INDEX	OPERATION
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230
231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280
281	282	283	284	285	286	287	288	289	290
291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310
311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370
371	372	373	374	375	376	377	378	379	380
381	382	383	384	385	386	387	388	389	390
391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410
411	412	413	414	415	416	417	418	419	420
421	422	423	424	425	426	427	428	429	430
431	432	433	434	435	436	437	438	439	440
441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460
461	462	463	464	465	466	467	468	469	470
471	472	473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488	489	490
491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510
511	512	513	514	515	516	517	518	519	520
521	522	523	524	525	526	527	528	529	530
531	532	533	534	535	536	537	538	539	540
541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560
561	562	563	564	565	566	567	568	569	570
571	572	573	574	575	576	577	578	579	580
581	582	583	584	585	586	587	588	589	590
591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610
611	612	613	614	615	616	617	618	619	620
621	622	623	624	625	626	627	628	629	630
631	632	633	634	635	636	637	638	639	640
641	642	643	644	645	646	647	648	649	650
651	652	653	654	655	656	657	658	659	660
661	662	663	664	665	666	667	668	669	670
671	672	673	674	675	676	677	678	679	680
681	682	683	684	685	686	687	688	689	690
691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710
711	712	713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728	729	730
731	732	733	734	735	736	737	738	739	740
741	742	743	744	745	746	747	748	749	750
751	752	753	754	755	756	757	758	759	760
761	762	763	764	765	766	767	768	769	770
771	772	773	774	775	776	777	778	779	780
781	782	783	784	785	786	787	788	789	790
791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810
811	812	813	814	815	816	817	818	819	820
821	822	823	824	825	826	827	828	829	830
831	832	833	834	835	836	837	838	839	840
841	842	843	844	845	846	847	848	849	850
851	852	853	854	855	856	857	858	859	860
861	862	863	864	865	866	867	868	869	870
871	872	873	874	875	876	877	878	879	880
881	882	883	884	885	886	887	888	889	890
891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910
911	912	913	914	915	916	917	918	919	920
921	922	923	924	925	926	927	928	929	930
931	932	933	934	935	936	937	938	939	940
941	942	943	944	945	946	947	948	949	950
951	952	953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968	969	970
971	972	973	974	975	976	977	978	979	980
981	982	983	984	985	986	987	988	989	990
991	992	993	994	995	996	997	998	999	1000
1001	1002	1003	1004	1005	1006	1007	1008	1009	1010
1011	1012	1013	1014	1015	1016	1017	1018	1019	1020
1021	1022	1023	1024	1025	1026	1027	1028	1029	1030
1031	1032	1033	1034	1035	1036	1037	1038	1039	1040
1041	1042	1043	1044	1045	1046	1047	1048	1049	1050
1051	1052	1053	1054	1055	1056	1057	1058	1059	1060
1061	1062	1063	1064	1065	1066	1067	1068	1069	1070
1071	1072	1073	1074	1075	1076	1077	1078	1079	1080
1081	1082	1083	1084	1085	1086	1087	1088	1089	1090
1091	1092	1093	1094	1095	1096	1097	1098	1099	1100
1101	1102	1103	1104	1105	1106	1107	1108	1109	1110
1111	1112	1113	1114	1115	1116	1117	1118	1119	1120
1121	1122	1123	1124	1125	1126	1127	1128	1129	1130
1131	1132	1133	1134	1135	1136	1137	1138	1139	1140
1141	1142	1143	1144	1145	1146	1147	1148	1149	1150
1151	1152	1153	1154	1155	1156	1157	1158	1159	1160
1161	1162	1163	1164	1165	1166	1167	1168	1169	1170
1171	1172	1173	1174	1175	1176	1177	1178	1179	1180
1181	1182	1183	1184	1185	1186	1187	1188	1189	1190
1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
1201	1202	1203	1204	1205	1206	1207	1208	1209	1210
1211	1212	1213	1214	1215	1216	1217	1218	1219	1220
1221	1222	1223	1224	1225	1226	1227	1228	1229	1230
1231	1232	1233	1234	1235	1236	1237	1238	1239	1240
1241	1242	1243	1244	1245	1246	1247	1248	1249	1250
1251	1252	1253	1254	1255	1256	1257	1258	1259	1260
1261	1262	1263	1264	1265	1266	1267	1268	1269	1270
1271	1272	1273	1274	1275	1276	1277	1278	1279	1280
1281	1282	1283	1284	1285	1286	1287	1288	1289	1290
1291	1292	1293	1294	1295	1296	1297	1298	1299	1300
1301	1302	1303	1304	1305	1306	1307	1308	1309	1310
1311	1312	1313	1314	1315	1316	1317	1318	1319	1320
1321	1322	1323	1324	1325	1326	1327	1328	1329	1330
1331	1332	1333	1334	1335	1336	1337	1338	1339	1340
1341	1342	1343	1344	1345	1346	1347	1348	1349	1350
1351	1352	1353	1354	1355	1356	1357	1358	1359	1360
1361	1362	1363	1364	1365	1366	1367	1368	1369	1370
1371	1372	1373	1374	1375	1376	1377	1378	1379	1380
1381	1382	1383	1384	1385	1386	1387	1388	1389	1390
1391	1392	1393	1394	1395	1396	1397	1398	1399	1400
1401	1402	1403	1404	1405	1406	1407	1408	1409	1410
1411	1412	1413	1414	1415	1416	1417	1418	1419</	

de l'"area", le schéma et le sous-schéma ont des liens avec les entités et les relations.

On remarque également que les groupes et éléments utilisent la même table. Cela se justifie quand on sait que les descriptions de l'un et l'autre ne présentent que d'infimes différences.

On observe aussi que le "MEMBER" de la relation voit ses attributs placés dans la table d'INDEX RELATION et d'INFO RELATION.

4.2.2. Canevas logique du SCHEMA. (voir figure 4.2)

La table DIRECTORY pointe dans l'INDEX SET sur l'enregistrement relatif au schéma. Cet enregistrement possède un pointeur vers l'INDEX SET à l'adresse des diverses areas mais aussi un pointeur vers l'adresse de début des enregistrements relatifs aux entités du schéma dans INDEX ENTITY et un pointeur vers l'adresse de début des enregistrements relatifs aux relations du schéma dans INDEX RELATION.

Dans chaque enregistrement relatif à une entité du schéma se trouve un pointeur vers l'adresse de début des groupes et éléments de cette entité dans la table d'INDEX GROUP-ELEMENT.

Bien entendu, par pointeur il faut entendre, outre l'adresse de début, l'indication du nombre d'enregistrements successifs à prendre en compte.

4.2.3. Canevas logique du SOUS-SCHEMA. (voir figure 4.2)

La table DIRECTORY pointe dans l'INDEX SET sur l'enregistrement relatif au premier sous-schéma. Les autres enregistrements relatifs aux sous-schémas suivent en séquence.

L'enchaînement est alors analogue à celui du schéma.

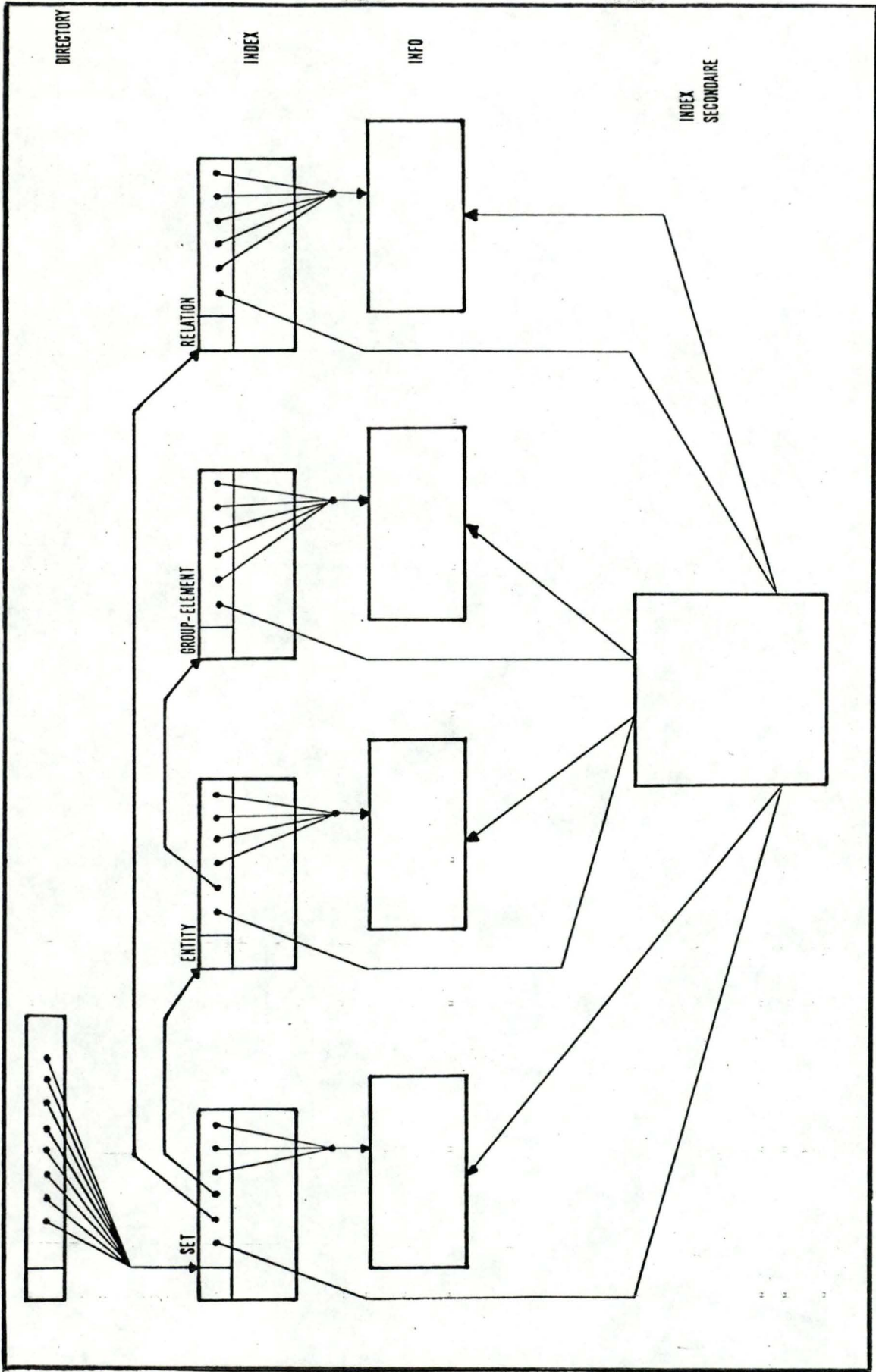


Figure 4.2

4.3. Remarque sur les tables.

Les responsables du système de documentation ont décidé pour plus de sécurité que le générateur ne pouvait accéder à la base de données DOCUMENTATION. L'information nécessaire à ce générateur doit alors se présenter ailleurs, dans des tables. On se trouve souvent dans l'impossibilité de quantifier le volume de tout ensemble d'information qui possède sa dynamique propre. De ce fait, les programmes doivent être recompilés pour effectuer les ajustements idoines.

Le système de tables adopté dans le système de documentation du CTISM rend le programme utilisateur indépendant des tables qu'il utilise. Un module de gestion des informations mémorisées, qui a pour nom PRVIRMAN, a été écrit par un responsable du système de documentation.

Pour lire, écrire ou agrandir l'espace d'une table (composée d'occurrences d'un type d'enregistrement unique) tout programme s'adresse à ce module spécialisé en lui fournissant divers paramètres.

Les tables sont chargées sur disques et gérées en mode mémoire virtuelle par ce module d'interface qui dispose d'un POOL de pages (= 1 Kbyte) en mémoire.

Ce module est d'une grande utilité dans le cas du générateur qui effectue énormément de lectures séquentielles, or les informations se trouvent dans le POOL en assez grand nombre et en séquence.

Chapitre 5 : Le générateur de descriptions de structure de base de données UDS/V2.

5.1. Entrée du générateur.

Le générateur retrouve dans les tables de rangement, évoquées au chapitre précédent, les valeurs des divers composants du schéma ou sous-schéma dont il va générer la description en UDS/V2.

5.2. Logique du générateur.

Le générateur, dont le schéma de travail est repris en figure 5.1., se déroule selon la séquence prévue pour toute description de base de données de type CODASYL, à savoir : SCHEMA ou SUB-SCHEMA, AREA, RECORD, GROUP ou ELEMENT, SET.

Il retrouve les valeurs de chacune des occurrences de chacun de ces objets grâce aux pointeurs décrits au paragraphe 4.2. du chapitre précédent en suivant l'enchaînement logique prévu.

Pour chaque clause de description à générer, il fait appel à un module spécialisé d'édition, une ou plusieurs fois, en lui transmettant des tables de valeurs propres à la clause. Ces valeurs sont de trois types :

Dés paramètres qui reprennent dans les tables de rangement l'information variable, c'est-à-dire des noms, des quantités,...

Des valeurs de conditions (vrai ou faux) qu'il calcule après la consultation des tables ; ces conditions portent sur la présence d'au moins une valeur pour un attribut, la présence de plusieurs valeurs pour cet attribut, une alternative dans le choix de la valeur, la fin des clauses d'un objet, ...

Des valeurs de positions qui indiquent à quelle distance de la marge on désire imprimer les informations de la ligne à générer.

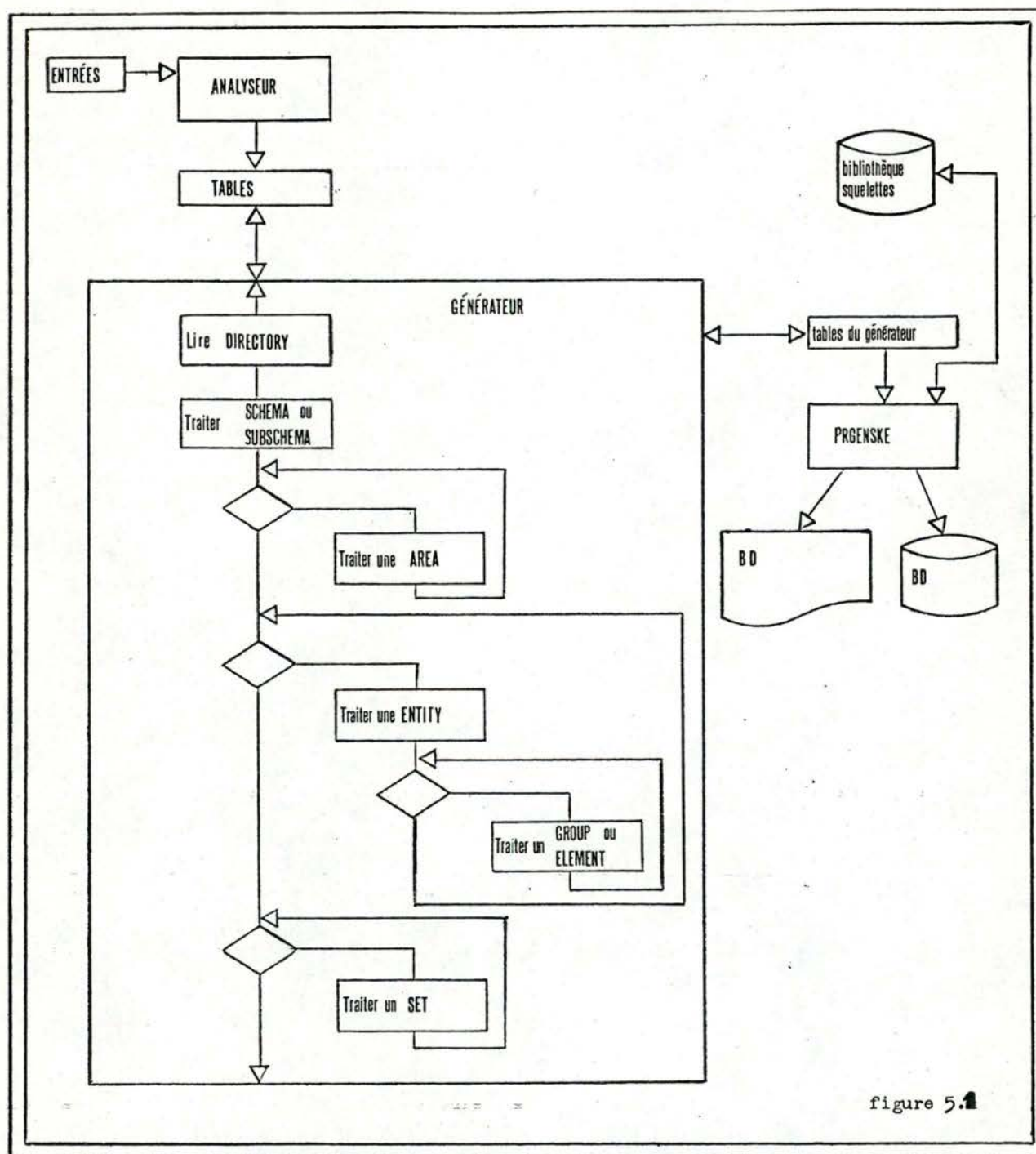


figure 5.1

5.3. Composition du générateur.

Le générateur est fait de trois programmes rédigés en COBOL.

Les organigrammes des programmes de génération sont repris aux pages A.4.1 à A.4.22.

Chaque programme est un générateur.

Le premier (PRDBGDDL) génère la description en langage DDL d'un schéma (voir A.5.1).

Le second (PRDBGSSL) génère la description d'un schéma en langage SSL (voir A.5.1); il s'agit d'un ajout propre à UDS/V2 qui comporte de multiples renseignements physiques sur le schéma. Le troisième programme (PRDBGSUB) génère la description d'un sous-schéma en langage DDL (voir A.5.2).

5.4. Module d'édition PRGENSKE.

Le module PRGENSKE, écrit par un responsable du CTISM, reçoit à chaque appel du générateur des tables de valeurs qu'il va utiliser pour générer les lignes d'impression prévues dans la description du squelette appelé.

5.4.1. Entrées du module PRGENSKE.

5.4.1.1. Squelettes descriptifs de la structure de base de données.

Le concept de squelette est une des outils principaux des générations automatiques du système de documentation du CTISM.

On constate que bien souvent le problème délicat réside dans la gestion de l'imprimé où une énorme partie est constituée par du texte qui doit être intégré et géré par le programme de génération, le rendant ainsi dépendant au plus haut point d'éventuelles modifications de la forme de sortie.

Le squelette est un moule qu'il faut garnir.

Il n'est pas sans rappeler le concept de "macro-instruction" où une liste de valeurs est fournie à chaque appel. Pour le squelette, il s'agit de tables de valeurs.

Forme de squelette

La décomposition hiérarchique d'un squelette est la suivante "SKELETON" composé de 1 à n "RECORD"

"RECORD" composé de 1 à n $\left\{ \begin{array}{c} \text{"PARAM"} \\ \text{ou} \\ \text{"TEXT"} \end{array} \right\}$

La forme complète d'un squelette est reprise en figure 5.2 ; c'est sous cette forme que toute la base de données UDS/V2 est décrite.

On trouvera la description complète des squelettes en annexe A.2.1. à A.2.8.

SKELETON

skeleton-name identifiant du squelette

NEW à la création du squelette

REPLACING (OLD) à la modification du squelette

RECORD caractérisé par

1. la possibilité de test d'une valeur de CONDITION dont dépend l'exécution du RECORD (si la condition est vérifiée, on prend en charge les PARAM et TEXT jusqu'au prochain RECORD)
2. Le passage au début de la ligne suivante de l'imprimé

condition-name nom de la condition à tester

IS $\left\{ \begin{array}{c} \text{TRUE} \\ \text{FALSE} \end{array} \right\}$ valeur(vraie ou fausse) à vérifier pour exécuter la génération indiquée

PARAM	
param-name	nom du paramètre à copier dans le texte généré
POS	position sur le support de sortie
PLUS	on compte le déplacement sur le support d'impression à partir de l'adresse où on s'est arrêté
integer	position fixe
V (name)	position variable (sur base de la valeur de la zone nommée)
TEXT	
text-inf	texte constant à copier dans la sortie générée.

Remarque : POS permet de gérer la disposition physique des mots de la ligne d'impression.

Cette description est compilée par un analyseur-chargeur qui en range le résultat dans la bibliothèque des squelettes.

Voici un exemple de description.

exemple : clause AREA NAME IS area-name
(AREA IS TEMPORARY)

```

SKELETON AREASK    NEW
  RECORD
    TEXT POS 12 VALUE AREA NAME IS
    PARAM POS PLUS 1 VALUE AREANM
  RECORD WHEN CONDITION TEMP IS TRUE
    TEXT POS POSIT VALUE AREA IS TEMPORARY

```

SKELETON *skeleton-name* $\left[\begin{array}{c} \{ \text{NEW} \\ \text{REPLACING} \end{array} \right] \left[\begin{array}{c} \text{OLD} \end{array} \right] \right]$

RECORD $\left[\begin{array}{c} \text{WHEN} \\ \{ \text{COND} \\ \text{CONDITION} \end{array} \right] \text{condition-name} \left[\begin{array}{c} \text{is} \\ \{ \text{TRUE} \\ \text{FALSE} \} \end{array} \right] \right]$

PARAM $\left[\begin{array}{c} \text{pos} \\ \{ \text{position} \} \end{array} \right] \left[\begin{array}{c} \text{PLUS} \\ \{ \text{integer} \\ \text{V (name)} \} \end{array} \right] \left[\begin{array}{c} \text{VALUE} \\ \text{parameter-name} \end{array} \right]$

TEXT $\left[\begin{array}{c} \text{pos} \\ \{ \text{position} \} \end{array} \right] \left[\begin{array}{c} \text{PLUS} \\ \{ \text{integer} \\ \text{V (name)} \} \end{array} \right] \left[\begin{array}{c} \text{VALUE} \\ \text{text-information} \end{array} \right]$

figure 5.2

5.4.1.2. Tables du générateur.
.....

Le générateur garnit ces tables et les transmet au module PRGENSKE en mentionnant le nom du squelette concerné.
Dans l'exemple du chapitre 5.4.1.1., les informations nécessaires sont :

nom du squelette	AREASK
condition	TEMP
paramètre	AREANM
position	POSIT

Table des paramètres (voir A.5.3.)

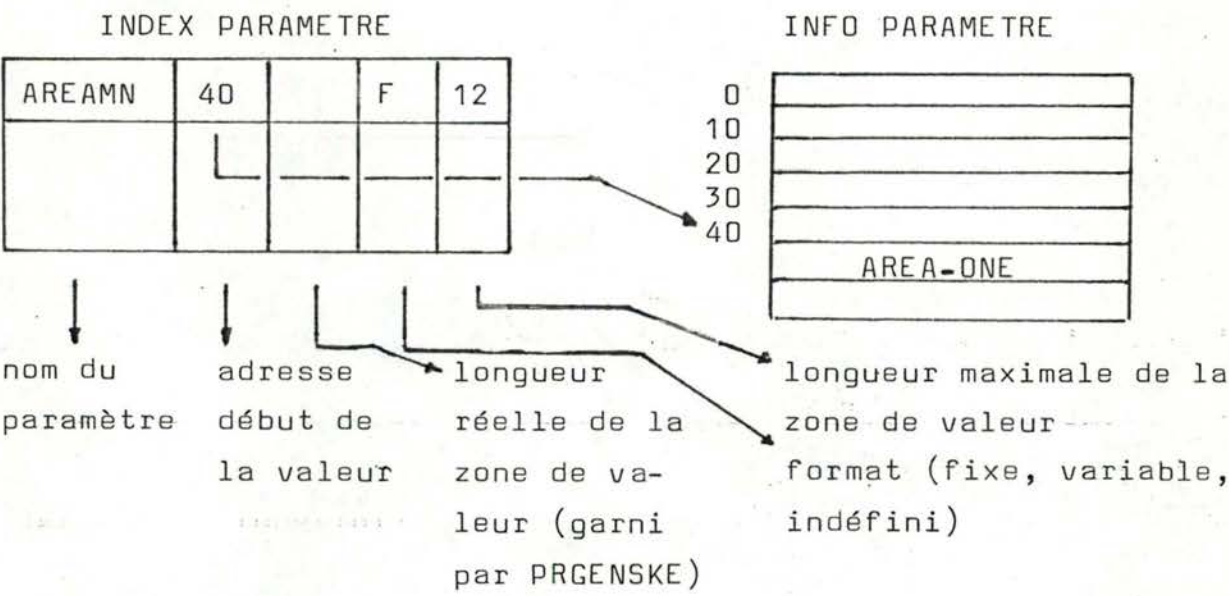


Table des conditions (voir A.5.4)

INFO CONDITION

TEMP	1

nom de la condition valeur de la condition

Table des positions (voir A.5.5)

INFO POSITION

POSIT	14

nom de la valeur de la
position position

5.4.2. Sortie du module PRGENSKE.

La forme de sortie de PRGENSKE suit la séquence de toute description CODASYL à savoir :

SCHEMA	SUB-SCHEMA
AREA	AREA
RECORD	RECORD
GROUP	GROUP
ELEMENT	ELEMENT
SET	SET

Comme le générateur, cette forme de sortie est constituée de trois blocs : pour le schéma en langage DDL, pour le schéma en langage SSL, pour le sous-schéma en langage DDL.

Cette forme de sortie est décrite en page A.3.1 à A.3.10.

On constate que non contente de reprendre les clauses de la base de données UDS/V2, elle offre des renseignements documentaires à titre de commentaires tels : courte description du contenu d'un objet, signification, nom du responsable de l'objet, numéro de version,...

5.4.3. Travail du module PRGENSKE.

PRGENSKE reçoit les quatre tables et le nom du squelette qu'il va extraire de sa bibliothèque de squelettes. En balayant les tables du générateur et le squelette, il garnit les valeurs variables (PARAM), effectue les tests (COND) et sort la clause décrite par le squelette.

Dans l'exemple précédent, le résultat est :

```

12
↓
AREA NAME IS AREA-ONE
  AREA IS TEMPORARY
    ↑
14

```

5.5. Ecriture des squelettes.

Procédé général de décomposition.

Dans la mesure du possible, chaque squelette est de taille raisonnable (< 1 K bytes) et couvre une portion de la sortie maximale à générer (voir A.3.1 à A.3.10).

exemple : entité du DDL ENTDDL
 titre de l'area du DDL AREDDL

Certaines clauses sont réunies en un seul squelette à des fins de clarté, elles sont alors mises sous condition.

On distingue 5 types de squelettes :

texte libre (sans condition), clause simple (sans condition), clause répétitive, clause à conditions imbriquées, clause sans condition.

Le nombre total de squelettes pour le générateur est de 46 unités.

Ces squelettes sont décrits en A.2.1. à A.2.8.

Problèmes de décomposition.

Les problèmes rencontrés lors de la décomposition (de la sortie maximale à générer) en squelettes sont dus à la syntaxe du squelette qui fait preuve d'un évident manque de souplesse dans le cas de certaines clauses de CODASYL et d'USD/V2.

Voici les 3 points délicats :

1° l'élément répétitif : on ne peut présager dans l'écriture du squelette du nombre de répétitions et donc du nombre d'occurrences de paramètres et/ou conditions avant l'exécution.

exemple: SEARCH KEY IS key-1 key-2
key-3...

2° les conditions imbriquées: un seul niveau de condition est possible à l'heure actuelle.

3° la position du point : la condition (portant sur la fin de clause, c'est-à-dire, l'écriture du . final) étant liée au RECORD réalise de ce fait l'avancement à la ligne suivante, le point est alors imprimé (s'il y a lieu) en-dessous de la dernière ligne de la clause

remarque : ce n'est qu'à l'exécution que l'on sait le nombre de clauses utilisées par un objet.

Des solutions ont été apportées, elles doivent être prises en compte par toute personne désireuse d'écrire des squelettes de description de structure de base de données dans le système de documentation du CTISM.

Voici les 3 solutions adoptées :

1° élément répétitif : le squelette décrit une occurrence de l'élément répétitif, il sera appelé un nombre de fois égal au nombre d'occurrences d'un élément répétitif.

remarque :-la première fois on imprime SEARCH KEY IS key-1
-pour tous les autres appels on positionne à 0 la condition régissant l'écriture des caractères SEARCH KEY IS (ce type de condition est toujours nommé IMP)

2° conditions imbriquées : le générateur réalise le test des différentes conditions et en garnit d'autres pour PRGENSKE

exemple : clause X
$$\left[\begin{array}{l} \text{info} \left[\begin{array}{l} P_3 = 0 \text{ info} \\ P_4 = 0 \text{ info} \end{array} \right] \\ P_2 = 1 \end{array} \right]$$

 $P_1 = 1$

devient $P_1 \rightarrow$ appel ou non du squelette

$P_2 \rightarrow$ rien

$P_3 \rightarrow P_3 = 0$ CONDITION1 = 1

$P_3 = 1$ CONDITION2 = 1

3° position du point : une solution mixte est proposée

a) pour les squelettes courts et simples une zone est garnie par le générateur.

exemple : TEXT POS 12 VALUE ZONE IS

PARAM POS PLUS 1 VALUE VALEUR

PARAM POS PLUS 1 VALUE POINT garni à . si fin de clause
 garni à b sinon

- b) pour les autres, le point sera lié à une condition et écrit sur la ligne suivante.

exemple : RECORD TEXT POS 10 VALUE ZONE IS

PARAM POS PLUS 1 VALUE VALEUR

RECORD WHEN COND END IS TRUE

TEXT POS 10 VALUE.

En règle générale, toute clause répétitive constitue un squelette tandis que toute clause présentant des alternatives devient un squelette où chaque branche de l'alternative est un RECORD mis sous condition.

5.6. Opérationnalité du générateur.

Le générateur est composé de trois programmes (dont les organigrammes se trouvent en annexe A4) rédigés en COBOL, compilés et rangés en librairie (± 3000 lignes).

La "WORKING STORAGE SECTION" où sont déclarées les zones et tables de travail ainsi que des tables du générateur est sensiblement identique pour les trois programmes.

Certaines procédures sont elles-aussi communes aux trois programmes (lecture des diverses tables..). De ce fait, la "WORKING-STORAGE SECTION" ainsi que ces procédures sont rangées en librairie et disponibles par COPY. Les squelettes sont définis et eux aussi mis en librairie.

Chapitre 6 : Etude comparative avec les logiciels SCAPFACE

et LEXICON

Les langages de description organique de base de données de ces deux systèmes de documentation sont basés sur les mêmes principes que celui de ce mémoire et sont analysés de manière analogue.

La comparaison porte sur le mode de rangement des informations du langage et surtout sur le principe de génération.

6.1 Le logiciel SCAPFACE.

PRESENTATION

SCAPFACE est un système de documentation développé dans le cadre du C.E.R.F.I.A. par Mr CHRISMENT et HUGET. La génération est l'oeuvre de Mr CHRISMENT et HENRY. Ils présentent un logiciel permettant, entre autres, la génération de structure de base de données dans tout langage de SGBD.

Cette génération s'effectue à partir de la description fonctionnelle du schéma de la base (enregistrée dans le dictionnaire de SCAPFACE) auquel le générateur intègre des caractéristiques propres à un SGBD précis grâce à un dialogue avec l'utilisateur.

Les informations sont classées en entités, rubriques, imprimés et applications. Elles sont stockées sous forme de listes chaînées.

La structure de base de données est représentée à l'aide d'une forêt qu'un algorithme de parcours par la méthode du pré-ordre et un algorithme d'insertion de parenthèses mettent sous forme parenthésée.

Le générateur estime, pour chaque ligne de la structure de base de données, tous les prédicats sur les valeurs présentes et positionne une et une seule condition (c_i). Cette condition entraîne l'exécution d'une procédure (d_i) de génération de ligne de structure de base de données.

CRITIQUE

L'apparente facilité de cette réalisation m'amène à faire quelques observations.

En admettant que l'utilisateur puisse introduire et mettre en librairie des informations précises sur le SGBD choisi, on est en droit de se poser deux questions.

Est-on limité aux langages de SGBD que l'on connaît et dont on possède les caractéristiques en librairie ?

Rentre-t-on des informations que l'on ne placera jamais dans le dictionnaire ?

L'alternative nous amène à penser que le dictionnaire n'est pas à jour ou que l'on se trouve aux antipodes du merveilleux outil présenté puisqu'on clôt le champ des langages de SGBD envisagés.

Néanmoins, la remarque principale se situe au niveau de la génération.

Chaque condition (établie sur base de combinaison de prédicats) sélectionne une procédure d'écriture de phrase du langage de SGBD choisi.

Pour toute phrase du langage de description organique de base de données présentant des options facultatives, il faut disposer d'autant de procédures d'écriture qu'il y a de combinaisons possibles d'options facultatives.

On imagine aisément l'éléphantinesque bibliothèque de programmes d'écriture qu'exige toute base de données de famille CODASYL.

Cette hyperdépendance à l'égard de la forme de sortie entraîne la recompilation des programmes à chaque modification.

Le logiciel SCAPFACE présente cependant des aspects positifs dans sa façon de ranger les informations en liste chaînée (gain d'espace, modification sans restructuration, ajout sans nouveau tri) et dans la simplicité de son dictionnaire où l'on ne retrouve que quatre types d'objets (entités, rubriques, imprimés, applications).

Conclusion

La méthode de génération actuelle de SCAPFACE ne semble pas être l'idéal. Elle ne permet que la traduction du canevas de structure de base de données et se limite empiriquement à ce seul objectif.

Il me semble qu'en toute logique, les promoteurs de ce projet vont s'orienter vers une solution du type de celle proposée ici. Ils ont en effet déjà reconnu dans leur présentation que "l'utilisation d'autres principes de génération basés sur la méthode des attributs sémantiques permet une prise en compte plus fine des caractéristiques spécifiques du SGBD".

6.2 Le logiciel LEXICON.

Présentation

LEXICON est un produit de la firme Arthur & Andersen. La description de base de données se fait par une entrée unique. Les informations sont rangées sous forme de listes chaînées. Peu de vérifications sont effectuées à l'analyse, les phrases sont considérées comme texte. L'équivalence de structure entre tout SGBD envisagé et le système de documentation LEXICON est réalisé par des phrases "TYPE IS".

La prise en charge des informations spécifiques aux bases de données se situe au niveau de la génération LEXICON qui est constituée de 4 phases (programmes).

Phase 1 = ANALYSE

Les composantes de la base de données sont retrouvées par sélection dans la liste chaînée des phrases et rangées en séquence dans une table.

Phase 2 : VALIDATION

Cette phase réalise le travail post-posé à l'introduction de la description. Elle analyse toutes les phrases de la table, en vérifie la cohérence et l'exactitude.

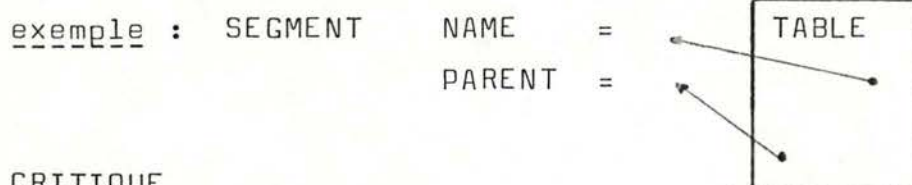
Phase 3 : TRADUCTION

Les mots clés initiaux du langage LEXICON sont remplacés par ceux du langage du SGBD choisi.

Phase 4 : GENERATION

Un générateur (spécifique au SGBD choisi) parcourt la table des

informations qu'il sort sous la forme correcte en ajoutant tous les mots réservés.



CRITIQUE

La description organique de base de données est prise en charge et rangée sous forme de listes chaînées sans être analysée. Différer la vérification présente de graves carences. D'une part, il n'est pas souhaitable que le dictionnaire contienne, à quelque moment que ce soit, des informations erronées. D'autre part, les erreurs sont détectées tardivement ce qui oblige l'utilisateur à effectuer ses corrections et à réintroduire ses phrases qui vont refaire tout le chemin jusqu'à la validation. Il me faut avouer que la solution présentée ici est entachée de ces mêmes inconvénients.

Tous les mots facultatifs des phrases sont rangés à la fois dans le dictionnaire et dans la table de génération alors qu'ils ne présentent aucun intérêt.

La construction d'une table reprenant en séquence la structure de base de données est d'un intérêt mitigé. Elle présente l'avantage de pouvoir disposer, à tout moment, d'un tout logique et d'un seul tenant.

La solution de ce mémoire possède le même esprit logique mais la séquence doit être retrouvée par des pointeurs. La structure des tables permet la consultation du tout mais aussi de chaque table, tandis que la structure des tables de LEXICON ne permet pas la consultation par un utilisateur intéressé par un type d'objet précis.

Le bât blesse également au niveau de la variabilité des entrées dans la table qui vont imposer l'emploi de tables à longueur

variable avec tout cela comporte de manipulations ou de tables fixes à longueur d'enregistrement maximale qui vont entraîner la réservation inutile de quantité d'espace mémoire. De plus, dans le premier cas, toute modification de la description organique de base de données impose la reconstruction d'une table ou du moins son reformatage.

La traduction des mots clés du langage LEXICON en langage du SGBD choisi est inutile. La correspondance peut être réalisée au niveau du générateur puisque celui-ci sait ce dont il a besoin.

<p><u>exemple</u> :</p> <p>NOM-LEXICON</p> <p>↓</p> <p>NAME</p>	<p>devient</p>	<p>GENERATEUR</p> <p>consultation de la table</p> <p>NAME ? si oui écrire NAME=</p>
-----------------------------------------------------------------	----------------	-------------------------------------------------------------------------------------

Le générateur pouvait rechercher "NOM-LEXICON" et le traduire en "NAME="

Le générateur reprend en séquence, par type d'objet, les clauses de sortie. Il regarde dans la table si le paramètre est présent. Si oui, il l'extrait, garnit la zone d'impression avec le mot réservé, y ajoute l'information de la table et imprime.

On déplore, une fois de plus, les inconvénients (flot d'opérations d'impression, dépendance vis à vis de la forme de sortie) qu'annihile le concept de squelette en délivrant le générateur de ces préoccupations pour le réduire au seul algorithme de parcours de schéma et sous-schéma avec garnissage de valeurs.

Les quatre phases de travail sont spécifiques à chaque type de SGBD.

La prise en compte d'un nouvel SGBD impose l'écriture de ces quatre phases. Dans la solution du mémoire, seul le générateur et les squelettes sont à écrire. La prise en compte d'un nouvel SGBD n'entraîne que l'ajout d'informations sans modification des structures de travail.

La remarque précédente est en grande partie due au fait que les responsables de LEXICON ont créé des objets spécifiques aux bases de données (DATABASE, SENSITIVITY, SUB-SCHEMA,...) alors qu'au CTISM on désire garder les objets "logiques" prévus par ISDOS (SET { schéma, sub-schéma, area, file } ,...)

Conclusion

La méthode de génération de LEXICON à l'encontre de SCAPFACE est commercialisée.

Une certaine liberté permet de multiples extensions pour des besoins futurs.

Des similitudes évidentes avec mon travail apparaissent :

1. structure des phrases du langage
2. rangement dans une table séquencée
3. analyse et validation spécifiques
4. traduction et génération spécifiques

Les phases 2 et 3 sont interverties dans ma solution et seule la phase 4 est complètement différente.

Il me faut honnêtement reconnaître que la première phase de mon travail s'est inspirée de LEXICON puisqu'elle a suivi ma rencontre avec Mr BONTINCKS de la firme ARTHUR & ANDERSEN à Bruxelles. Les autres phases de LEXICON ne m'ont été révélées qu'après l'accomplissement de mon travail par Mr VERGNANGEAL de la même firme à Paris.

6.3. Améliorations techniques

Il n'est pas envisagé ici de modifications dues aux leçons des logiciels SCAPFACE et LEXICON.

Le problème étudié en ce mémoire a été résolu dans une optique de clarté et dans le respect de normes établies. Néanmoins, certaines améliorations ponctuelles sont envisageables dans l'immédiat.

1. Tables d'index

Une place maximale égale à la somme des longueurs des informations différentes possibles pour les SGBD envisagés a été réservée. On peut limiter cet espace à la dimension maximale des informations du SGBD le plus gourmand. Cet espace est formaté différemment pour chaque SGBD ou laissé en format libre (ce qui oblige, pour retrouver le pointeur d'un numéro d'attribut, à adjoindre à ce pointeur le numéro de l'attribut et nécessite la recherche par SEARCH).

2. Découpe des phrases en attributs

On peut grouper des attributs pour en diminuer le nombre et gagner de la place.

$$\begin{array}{l}
 \text{exemple : } \left\{ \begin{array}{l} \text{MANDATORY} \\ \text{OPTIONNAL} \end{array} \right\} \left\{ \begin{array}{l} \text{AUTOMATIC} \\ \text{MANUAL} \end{array} \right\} \\
 \quad \quad \quad 1 \quad \quad + \quad \quad 1 \\
 \quad \quad \quad = 2 \text{ attributs}
 \end{array}
 \Rightarrow
 \begin{array}{c}
 \begin{array}{ccccc}
 & & & & \text{valeur} \\
 \left\{ \begin{array}{l} M \quad A \\ M \quad M \\ O \quad A \\ O \quad M \end{array} \right. & \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} & \left. \vphantom{\begin{array}{l} M \quad A \\ M \quad M \\ O \quad A \\ O \quad M \end{array}} \right\} & \begin{array}{l} 1 \text{ attri-} \\ \text{but} \end{array}
 \end{array}
 \end{array}$$

On peut de même codifier les phrases pour n'en faire qu'un attribut voire 2 ou 3.

3. Rangement des informations codées

Jusqu'à présent, quelques informations sont codées. Si l'on envisage l'amélioration n° 2, on codifiera toutes les informations. Ces codes sont présentement référencés par un pointeur (dans INDEX) et placés dans une zone fixe (dans INFO). On peut déplacer ce code dans INDEX puisqu'il peut tenir dans la zone réservée au pointeur. L'utilisateur sait que pour telle position de l'INDEX il doit considérer un code et non plus un pointeur.

Conclusion

Ce mémoire propose une réalisation de génération automatique de descriptions de structure de base de données dans le système de documentation du CTSM.

L'utilisateur définit sa base de données dans le langage de description organique de base de données.

L'analyseur (spécifique) traite les phrases du langage dont il range l'information dans des tables disponibles à tout instant pour l'utilisateur et aisées à l'emploi.

Le générateur extrait de ces tables l'information sur la base de données. Il puise le moule descriptif de la forme des clauses de sortie du SGBD UDS/V2 dans la bibliothèque des squelettes. Il fusionne ces deux sources de renseignement afin de générer le schéma ou le sous-schéma UDS/V2.

Le langage englobe les exigences du SGBD UDS/V2 et de la norme CODASYL, tandis que le générateur est spécifique au SGBD UDS/V2.

La spécificité des programmes de génération est telle qu'ils ne peuvent être exécutés que dans le contexte du système de documentation en projet.

En effet, le générateur use de deux interfaces avec des modules spécialisés (PRVIRMAN : gestion des tables en mode virtuel, PRGENSKE : gestion des squelettes) et nécessite l'écriture préalable de l'analyseur et l'analyseur spécifique.

Or, tous ces éléments sont à l'état de projet et ne pourront être opérationnels qu'à l'arrivée du matériel informatique (prévu) et à la mise en place de saines conditions de travail.

La tendance générale dans les centres informatiques est d'automatiser le maximum (raisonnable) de traitements afin d'éviter le risque d'erreurs humaines et atténuer le coût pro-

hibitif du personnel. Le concept de système de documentation abonde dans ce sens en permettant, en plus de la documentation, des automatismes pré-programmés comme le sujet de ce mémoire. La plupart des systèmes de documentation (ou dictionnaires de données) incluent à leur éventail de "facilités" ce mécanisme de génération.

Il est à regretter que le temps imparti et les impératifs du CTISM n'aient pas permis d'élaborer une solution plus générale, plus théorique.

A ce propos, il conviendrait d'étudier le concept de squelettes afin d'en étendre les possibilités et d'élaborer une proposition de pseudo-code afin d'envisager la possibilité de génération automatique d'instructions en langage de programmation évolué.

Quoi qu'il en soit, le contact d'un personnel informatique compétent et l'apprentissage d'un système de documentation auront été pour moi un enrichissement de tous les jours.

Références

(1) Norme CODASYL DBTG Avril 1971

(2) Software product UDS/Universal Data Base Management System

UDS/V1	DDL	Juin 1976
"	SSL SCHEMA	" "
"	SUB-SCHEMA	" "
UDS/V2	DDL SCHEMA	Juillet 1978
"	SSL	" "
"	SUB-SCHEMA	" "

(3) Entretiens (+ documents) avec Mr BONTINCKS (firme ARTHUR & ANDERSEN) Bruxelles 56, avenue des Arts Août 1978

(4) Entretien (+ documents) avec Mr VERGNANGEAL (firme ARTHUR & ANDERSEN) Paris Tour Gan (40°) La Défense 2
Janvier 1979

(5) Conférences sur les dictionnaires de données

"Le logiciel CULLINAM" par M. GOETGELUCK (Prof K.U.L.)
INSTITUT D'INFORMATIQUE 21 octobre 1978

"Le logiciel ISDOS" par M.F. BODART (Prof F.N.D.P.)
M.Y. PIGNEUR (Assistant F.N.D.P.)
INSTITUT D'INFORMATIQUE 18 novembre
1978

"Le logiciel LEXICON" par M.P. BONTINCK (Arthur ANDERSEN
and Co)
INSTITUT D'INFORMATIQUE 16 décembre
1978

"Le logiciel MINOS III" par M. THULY (PEUGEOT-FRANCE)
INSTITUT D'INFORMATIQUE 24 mars 1979

"Synthèses et conclusions sur les dictionnaires de données"
par F. BODART (Prof F.N.D.P.) INSTITUT D'INFORMATIQUE
28 avril 1979

- (6) Périodique "OI INFORMATIQUE"
Banc d'essai pour 8 dictionnaires de données Novembre 1978
- (7) SOBEMAP : 44^e journée d'informatique : SYSTEM DICTIONNARY
1978
- (8) Mémoire de H. ZIMMER Année académique 1976-1977
Contenus et usages des dictionnaires de données
- (9) Mémoire de Y. PIGNEUR Année académique 1976-1977
Synthèses et propositions de spécifications théoriques
d'un dictionnaire de données.
- (10) Bulletin de liaison du club banques de données
Projet SCAPFACE : conception d'un SGBD
MM ANTON, CHRISMENT, CRAMPES, LUGUET n° 12 Décembre 1973
Projet SCAPFACE : accès à l'information
MM ANTON, CHRISMENT, CRAMPES n° 15 Juillet 1975
Projet SCAPFACE : génération de structures de données
MM CHRISMENT, HENRY n° 17 Mai 1978
- (11) Manuel ISDOS version 4.2.
- (12) PSL : language reference summary (ISDOS PROJECT)
UNIVERSITY OF MICHIGAN Novembre 1977

A N N E X E S

- A1 Langage de description organique de base de données.
- A2 Description des squelettes.
- A3 Sortie générée.
- A4 Organigrammes des programmes de génération.
- A5 Divers : Schéma DDL
 - Schéma SSL
 - Sous-schéma DDL
 - Table des paramètres de génération
 - Table des conditions de génération
 - Table des positions de génération

LANGAGE DE DESCRIPTION ORGANIQUE DE BASE DE DONNEESLISTE DES PHRASES

(ajouts pour la génération de structure de base de données)

Légende

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

au moins 1

au plus 1

$$\begin{pmatrix} | \\ | \\ | \end{pmatrix}$$
occurrence de a ou
b ou c
$$\begin{pmatrix} | & a & | \\ | & b & | \\ | & c & | \end{pmatrix}$$

au moins 1

au plus 1 de chaque

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

au moins 0

au plus 1

- - -

répétition permise

MOT

mot facultatif

MOT

" réservé obligatoire

mot

opérande (numérique, alphabétique, alphanumérique)

Remarques

TYPE

type de la phrase ou de l'attribut

O (OBLIGATOIRE)

F (FACULTATIF)

FOR ou ON

pour un objet différent de celui où est
écrite cette phrase

IN

dans un autre objet (ex.: IN sous-schéma)
(ces 2 attributs relèvent de l'analyseur, ils
lui permettent de replacer les autres attributs
de la phrase dans la description de l'objet
idone et pour le set choisi).

Pour plus de précisions sur le sens exact des mots (réservés ou autres)
l'utilisateur est prié de se référer à la brochure du SGDB choisi.

CHECK

ON {element}
{group}-name OF version-number IS

PICTURE
RANGE OF literal THRU literal
 proc-name USING {element}-name [, {element}-name] ---
 {group}

Définition

inhibe la conversion des données ou spécifie une procédure de contrôle de validité à exécuter lorsqu'une valeur est changée ou ajoutée à la **B.D.**

<u>DECRI</u>	<u>DANS</u>	ENTITY	<u>OCCURRENCE</u>	0 - N
	POUR	ELEMENT-GROUP	"	0 - 1

TYPE FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
----	----------	------	---------------

* ON ____

301	0	type de check
302	0	littéral du check
303	0	procédure du check
304	0	donnée pour le check

REGLES

N° ATTRIBUT

```

301      valeurs possibles : 1, 2 , 3
302      au nombre de 2 obligatoirement
303      entraîne la présence d'au moins
304      1 attribut 304
305      entraîne la présence obligatoire
306      d'1 attribut 303

```


COMPRESSIONFOR ALL ITEMS

<u>DEFINITION</u>	réalise la compression physique des éléments d'une entité		
<u>DECRIE DANS</u>	ENTITY		
<u>OCCURRENCE</u>	0 - 1		
<u>TYPE</u>	FACULTATIVE		
<u>DECOMPOSITION</u>	<u>N°</u>	<u>ATTRIBUT</u>	<u>TYPE</u> <u>SIGNIFICATION</u>

305	o	présence ou absence de la phrase
-----	---	----------------------------------

CONDITION

FOR {element}
 { group } -name OF version-number IN SET set-name

IS string {VALUE IS } litteral [{THRU } litteral]
 {VALUES ARE } [{THROUGH }]
 [,litteral [{THRU } litteral]] ---
 [{THROUGH }]

<u>DEFINITION</u>	renomme l'élément décrit en fonction de sa valeur et, pour ce faire, associe un nom à un ensemble de valeurs ou à une valeur unique.		
<u>DECRIE DANS</u>	ENTITY		
<u>POUR</u>	ELEMENT DANS UN SOUS-SCHEMA		
<u>OCCURRENCE</u>	0 - N		
<u>TYPE</u>	FACULTATIVE		
<u>DECOMPOSITION</u>	<u>N°</u>	<u>ATTRIBUT</u>	<u>TYPE</u> <u>SIGNIFICATION</u>

*

FOR ...

306	O	nom de condition
308	F	littéral début fourche de valeurs ou valeur
309	F	littéral fin fourche de valeurs

REGLESN° ATTRIBUT

308	littéral numérique
309	" "

CONVERSIONON {element}
 {group} -name OF version-numberFOR {ENCODING}
 {DECODING} [ALWAYS] CALL process-nameDEFINITION

procédure à exécuter lorsqu'un élément, exigeant une conversion spéciale, est extrait ou mis à jour.

DECRIE DANS
POURENTITY OCCURRENCE 0 - N
ELEMENT-GROUP " 0 - 1TYPE

FACULTATIVE

DECOMPOSITION

N° ATTRIBUT TYPE SIGNIFICATION

✱

ON ...

310

O

type de conversion

311

F

appel dans tous les cas ou pas

312

O

nom procédure

REGLES

N° ATTRIBUT

310

valeurs possibles : 1 , 2

DBKEYTRLTB

[IS integer] WITHIN AREA set-name [version-number]

DEFINITION

spécifie dans quel "set (area)" sera placée la table de traduction des data-base-key et quel espace elle occupera.

DECRIE DANS

ENTITY

OCCURRENCE

0 - 1

DECOMPOSITION

N° ATTRIBUT TYPE SIGNIFICATION

313

F

nom area

314

F

nombre de pages

DBPRIVACY

[ON { element
group
entity
relation
set (area)
set (S-SCH) } -name OF version-number [IN SET set-name]]

[FOR OPERATION [ALTER
COMPILE
COPY
DISPLAY
FIND
GET
INSERT
LOCK
MODIFY
ORDER
REMOVE
STORE
[EXCLUSIVE (RETRIEVAL)
PROTECTED (UPDATE)]]]

IS SPECIFIED BY { PROCEDURE proc-name } [OR { PROCEDURE proc-name }
{ string } { string }]

DEFINITION

spécifie des verrous pour certaines opérations sur cet objet;
à l'exécution, des clés seront nécessaires pour effectuer ces
opérations sur ces objets.

DECRIE DANS

SET
ENTITY
RELATION

POUR

ELEMENT, GROUP, ENTITY, RELATION, AREA, SOUS-SCHEMA, SCHEMA

OCCURRENCE

0 - N

TYPE

FACULTATIVE

DECOMPOSITION

N° ATTRIBUT TYPE SIGNIFICATION

* 1 F ON ...
* 2 F IN ...

315 F type d'opération
316 F type d'opération (si l'intitulé est en
2 parties)
317 O type de verrou
318 O verrou

REGLES

N° ATTRIBUT

316 entraîne la présence de 317
317 valeurs possibles : 1, 2

* 1 pour schéma
* 1+* 2 pour sous-schéma

INDEX

string IS { ATTACHED TO OWNER
 { DETACHED WITHIN AREA set-name [version-number] } }

[AND TYPE IS { DB-KEY-LIST
 { REPEATED- KEY }]

DEFINITION

crée et ordonne un ensemble de renseignements qui permettront de retrouver une entité sur base de certaines de ses clés; spécifie comment il sera construit et où il sera placé.

DECRIE DANS

ENTITY
 RELATION

OCCURRENCE

0 - N

TYPE

FACULTATIVE

REGLES DE LA PHRASE

est entraînée par la présence d'un index dans le SEARCH ;
 entraîne éventuellement la présence d'une phrase
 REORGANIZATION

DECOMPOSITION

N° ATTRIBUT TYPE SIGNIFICATION

319	O	nom de l'index
320	F	placement de l'index
321	F	nom de l'area de placement
322	F	type d'index

REGLES

N° ATTRIBUT

322 valeurs possibles : 1, 2
 1 = DATABASE-KEY-LIST

KEY

FOR { ASCENDING } [RANGE]
 { DESCENDING }

IS {element}-name [version-number] [{element}-name] - - -
 {groupe}

[DUPLICATES ARE [FIRST]
 [LAST] ALLOWED]
 [NOT]]

DEFINITION

spécifie les clés et l'ordre de tri pour un member de relation et règle le cas des members qui présentent les mêmes ensembles de valeurs de clés.

DECRIE DANS

RELATION

POUR

ENTITY (MEMBER)

OCCURRENCE

0 - 1 , 0 - N (pour CODASYL)

TYPE

FACULTATIVE

DECOMPOSITION

N° ATTRIBUT	TYPE	SIGNIFICATION
323	O	type de clé
324	F	règles spéciales
325	O	nom de clé
326	F	type de duplication

REGLESN° ATTRIBUT

323	valeurs possibles : 1 , 2
324	— ordre des clés va du majeur au mineur
	— clés ascendantes et descendantes sont mélangées
	— si pas de clé, la d-b-key est utilisée en clé ascendante
	— clés des ≠ members doivent être de format similaire
	— ≠ members triés sont rangés séquentiellement
	— sa présence entraîne le respect des règles décrites ci-dessus
326	valeurs possibles : 1, 2, 3

LOCATION

MODE IS

```

{ -DIRECT {element} -name [version number]
  { group }
- CALC [process-name] USING {element} -name [version-number]
  { group }
  {element} -name
  { group }
  --- DUPLICATES ARE [NOT] ALLOWED
- VIA relation-name [version-number] RELATION

```

DEFINITION

donne le critère de sélection d'une occurrence d'une entité et le critère de placement de cette même occurrence au sein d'un "set (area)".

DECRIE DANS

ENTITY

OCCURRENCE

0 - 1

TYPE

FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
327		O	type de location mode
328		F	procédure pour la recherche par CALC
329		F	nom de relation (recherche par VIA)
330		F	" " clé (recherche par DIRECT)
331		F	" " " (recherche par CALC)
332		F	type de duplication

REGLES

N° ATTRIBUT

327

valeurs possibles : 1, 2, 3

328

entraîne la présence d'au moins un attribut : 331, 332

331-332

obligatoire si l'attribut 327 a la valeur 2

332

valeurs possibles : 0, 1

MEMBEROF RELATION IS entity-name [version-number]DEFINITION

nomme un membre d'une relation

DECRIE DANS

RELATION

OCCURRENCE

1 , 1 - N (pour CODASYL)

TYPE

OBLIGATOIRE

REGLES POUR LA PHRASE

1. entraîne la présence obligatoire de TYPEAPPART
2. entraîne la présence facultative de KEY, SEARCH, SELECTION
3. précède les phrases TYPEAPPART, KEY, SEARCH, SELECTION pour CE member

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
333		O	nom de l'entity member

MODE

IS

```

{ -CHAIN [LINKED TO PRIOR]
  { {POINTER-ARRAY [DYNAMIC]}
    {LIST
      {
        { ATTACHED TO OWNER
          DETACHED [WITHIN AREA set-name [version-number] ]
        }
        [ WITH PHYSICAL LINK ]
      }
    }
  }
}

```

<u>DEFINITION</u>	spécifie le mécanisme utilisé pour réaliser les manipulations de la relation		
<u>DECrit DANS</u>	RELATION		
<u>OCCURRENCE</u>	0 - 1	(pour UDS)	1 (pour CODASYL)
<u>TYPE</u>	FACULTATIVE	"	OBLIGATOIRE "
<u>DECOMPOSITION</u>	N°	ATTRIBUT	TYPE SIGNIFICATION
	334	O	type de mode
	335	F	lié à l'owner ?
	336	F	dynamic ? (contrôle à l'exécution (DML))
	337	F	placement du mode
	338	F	nom de l'area où on placera le mode

REGLESN° ATTRIBUT

334	valeurs possibles : 1, 2, 3
335-336	présence entraîne la véracité de la signification
337	valeurs possibles : 1, 2, 3

OCCURS

FOR {element} -name OF version-number [IN SET set-name]

IS integer [TO integer] TIMES [DEPENDING ON {element} -name]

[INDEXED BY string [,string] ---]

DEFINITION définit un vecteur ou un groupe répétitif

DECRIE DANS ENTITY

POUR ELEMENT-GROUP dans un sous-schéma

OCCURRENCE 0 - N

TYPE FACULTATIVE

<u>DECOMPOSITION</u>	N° ATTRIBUT	TYPE	SIGNIFICATION
	1	O	FOR ...
	2	O	IN ...
	339	O	nombre d'occurrences (ou début d'une fourche de valeur)
	340	F	nombre d'occurrences (limite max de la fourche)
	341	F	nom element
	342	F	" d'index

<u>REGLES</u>	N° ATTRIBUT	
	339-340	bornes de valeur (ex: 1 TO 12)
	341	nom élément dont dépend la longueur de la fourche
	342	index COBOL pour balayer la table
	340	entraîne la présence obligatoire d'l attribut 341
	341	" " " "
		attribut 340

REMARQUE pour le schéma, le facteur répétitif est extrait de la relation
 [CONSISTS OF n groupe [] --] element [] --] de l'objet
 entity et/ou groupe.

OPERATION

[ON {element}
 { group } -name OF version-number [FOR SET set-name]

FOR

DELETE
DELETE ALL
DELETE ONLY
DELETE SELECTIVE
FIND
GET
INSERT
MODIFY
ORDER
REMOVE
STORE
OPEN

FOR

EXCLUSIVE

PROTECTED

NON-EXCLUSIVE

UPDATE

RETRIEVAL

EXCLUSIVE

PROTECTED

NON-EXCLUSIVE

CLOSE

CALL process-name

[USING

{element}
{ group }

-name

[version-number]

[, {elem}
{ gr } -name]DEFINITION

spécifie les procédures à exécuter lorsque certaines commandes sont effectuées sur certains objets.

DECRIE PAR

ENTITY
RELATION
SET

POUR

ENTITY, RELATION, ELEMENT, GROUPE, AREA (pour CODASYL)

OCCURRENCE

0 - N

TYPE

FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
*1		O	ON...
*2		O	FOR...
343		F	type d'opération (partie 1)
344		F	type d'opération (partie 2)
345		F	type d'opération (partie 3)
346		O	nom de la procédure
347		F	nom de variables pour la procédure

REGLES

N°	ATTRIBUT	REGLE
*1		si l'OPERATION se rapporte à un élément
*2		si l'OPERATION se rapporte à une entity d'une DB bien précise

ORDER

IS (ALWAYS (FIRST
LAST
NEXT
PRIOR
IMMATERIAL))
SORTED INDEXED NAME IS string WITHIN RECORD-NAME
BY DATABASE-KEY
BY DEFINED KEYS DUPLICATES
ARE FIRST ALLOWED
LAST
NOT)

DEFINITION spécifie le point d'insertion d'une occurrence d'1 entity member dans 1 relation et ainsi définit l'ordre de progression séquentielle

DECRIE PAR RELATION

OCCURRENCE 1

TYPE OBLIGATOIRE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
348		F	always ?
349		O	type d'ordre
350		F	type de sorted de 1'ordre
351		F	indexed ?
352		F	nom de 1'index
354		F	type de duplication

REGLESN° ATTRIBUT

348	si oui, l'effet de la commande ORDER du DML restera local au programme
349	valeurs possibles : 1,2,3,4,5,6
350	valeurs possibles : 1,2,3
352	entraîne la présence d'1 phrase INDEX
354	valeurs possibles : 0,1,2,3

PICTURE

FOR {element} -name OF version-number IN SET set-name IS string
 { group }

DEFINITION décrit les caractéristiques d'un élément (format physique)

DECRIE DANS ENTITY OCCURRENCE 0 - 1
POUR ELEMENT dans SOUS-SCHEMA " 0 - N

TYPE FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
* 1		O	FOR...
* 2		O	ON...
355		O	caractères de description (ex: x(30))

REGLEN° ATTRIBUT

355	pour le sous-schéma, il est donné, tandis que pour le schéma, il faut le construire en allant quérir les différentes clauses physiques : LENGTH, INTERNAL-FORM, ...
-----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------

PLACEMENT

OPTIMIZATION FOR RELATION relation-name [version-number]

DEFINITION spécifie pour quelle relation (à laquelle participe cette entité) il convient d'optimiser le placement

DECRIE DANS ENTITY

OCCURRENCE 0 - 1

TYPE FACULTATIF

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	356	0	nom de relation

POPULATION

IS integer [, INCREASE IS integer] WITHIN set-name [version-number]
 [, WITHIN set-name IS integer]

DEFINITION détermine le nombre d'objets, leur augmentation prévisible ainsi que leur répartition par "set (area)"

<u>DECRIE DANS</u>	ENTITY	<u>OCCURRENCE</u>	0 - N
	RELATION	"	0 - 1

TYPE FACULTATIVE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	357	0	nombre d'objets présents
	358	F	augmentation prévisible du nombre d'objets
	359	F	nom area

<u>REGLES</u>	<u>N° ATTRIBUT</u>	
	357 - 359	pour l'entity
	357 - 358	pour la relation

PRIVACY KEY

FOR COPY IS {litteral}
 { string }

DEFINITION spécifie la clé d'accès à un "set (schéma)" qui a prévu un verrou pour l'opération de création de "set(sous-schéma)" (= FOR COPY)

DECRIE DANS SET
POUR SOUS-SCHEMA

OCCURRENCE 0 - 1

TYPE FACULTATIVE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	360	0	clé

<u>REGLE</u>	<u>N° ATTRIBUT</u>	
	360	valeur ou nom de clé

RENAMING

[AREA set-name [version-number] IN SCHEMA IS CHANGED TO set-name]---

[ENTITY entity-name [version-number] IN SCHEMA IS CHANGED TO entity-name]---

[ELEMENT elem-name [version-number] IN SCHEMA IS CHANGED TO elem-name]---

[GROUP group-name [version-number] IN SCHEMA IS CHANGED TO group-name]---

[RELATION relation-name [version-number] IN SCHEMA IS CHANGED TO relation-name]---

[IMPLEMENTOR impl-name [version-number] IN SCHEMA IS CHANGED TO impl-name]---

DEFINITION donne à certains objets du schéma un nouveau nom lors de l'emploi de ces objets dans un sous-schéma : ceci, à des fins de convention nominale de langage hôte ou pour faciliter certains traitements

DECRIE DANS SET
POUR ELEMENT, GROUP, ENTITY, RELATION, AREA dans un SOUS-SCHEMA

OCCURRENCE 0 - N

TYPE FACULTATIVE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	363	0	type d'objet
	364	0	nom de l'objet dans schéma
	365	0	nouveau nom de l'objet pour le sous-schéma

REORGANIZATIONDYNAMIC SPANS integer PAGES [FOR string]DEFINITION spécifie le nombre de pages impliquées dans la réorganisationDECRIE DANS ENTITY
RELATIONOCCURRENCE 0 - 1TYPE FACULTATIVE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	361	O	nombre de pages
	362	F	nom d'index

REGLE N° ATTRIBUT

362 la phrase concerne un index déjà défini dans SEARCH ou ORDER du DDL et dans INDEX du SSL, l'analyseur devra intégrer ce renseignement à ceux de cet index

RESTRICTIONFOR set-name OF version-nbr CONCERN [RELATION relation-name]

[,relation-name] --- [AREA set-name [, set-name] ---]

DEFINITION limite pour un "set (sous-schéma)" le nombre de relation et de "set (area)" prévus dans le "set (schéma)"DECRIE DANS ENTITY
POUR RELATION et AREA dans un SOUS-SCHEMAOCCURRENCE 0 - 1TYPE OBLIGATOIRE (s'il y a lieu de restreindre)

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	* 1	O	FOR
	* 2	F	nom relation
	* 3	F	nom area

REGLE N° ATTRIBUT

* 2-* 3 l'analyseur, découvrant dans l'entity cette phrase, supprimera de ses tables de sortie les areas et relations mises en cause

RESULT

FOR element-name OF version-number IS $\left\{ \begin{array}{c} \text{ACTUAL} \\ \text{VIRTUAL} \end{array} \right\}$ OF proces-name

$\left[\begin{array}{c} \text{USING} \left\{ \begin{array}{c} \text{element} \\ \text{group} \end{array} \right\} \text{-name} \left[\begin{array}{c} \text{,} \left\{ \begin{array}{c} \text{element} \\ \text{group} \end{array} \right\} \text{-name} \end{array} \right] \end{array} \right]$

$\left[\begin{array}{c} \text{ON MEMBERS entity-name} \left[\begin{array}{c} \text{, entity-name} \end{array} \right] \text{--- OF RELATION relation-name} \end{array} \right]$

DEFINITION

spécifie une procédure à exécuter et le fait que le résultat doit ou ne doit pas être physiquement contenu dans l'entity

DECRIE DANS
POUR

ENTITY	<u>OCCURRENCE</u>	0 - N
ELEMENT	"	0 - 1

TYPE

FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
----	----------	------	---------------

*		0	FOR...
366		0	type de placement du résultat
367		0	nom de procédure
368		F	paramètres pour la procédure
369		F	noms des membres
370		F	nom relation

REGLE

N°	ATTRIBUT
----	----------

369-370

l'élément doit être l'owner de l'attribut 370 et les membres doivent être les attributs 369

SEARCH

KEY IS {element}
 {group}-name [version-number] [, {element}
 {group}-name]-----

[
USING {CALC
 {INDEX [NAME IS string]}
 (process-name)}
]

DUPLICATES ARE NOT ALLOWED

DEFINITION

déclare au SGBD qu'il va lui falloir créer un index pour toutes ses entités membres, précise les clés à consulter et spécifie le type de duplication à observer

DECRIE DANS

ENTITY
RELATION

OCCURRENCE

0 - N

TYPE

FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
----	----------	------	---------------

371	O	nom de clé
372	F	type de search
373	F	nom de procédure
374	F	nom d'index
375	O	type de duplication

REGLES

N° ATTRIBUT

372 obligatoire en UDS (valeurs : 1,2,3)
374 entraîne la présence d'une phrase
INDEX

SELECTION

[MEMBER entity-name [version-number] IN SET set-name]

IS THRU [relation-name [version-number] USING]

CURRENT OF SET

LOCATION MODE OF OWNER

[[USING {element} {group}-name [version-number] , {element} {group}-name]]]
[[ALIAS FOR {element} {group}-name [version-number] IS string]]]

relation-name [version-number] (USING {element} {group}-name [version-number] , {element} {group}-name)
(ALIAS FOR {element} {group}-name IS string)

DEFINITION donne les règles de sélection de l'occurrence de relation idoine, dans le but d'insérer ou d'accéder à un membre-entité choisi

DECRIE DANS RELATION
POUR ENTITY (MEMBER) (dans SOUS-SCHEMA)

TYPE FACULTATIVE

DECOMPOSITION	N°	ATTRIBUT	TYPE	SIGNIFICATION
	*	1	F	MEMBER...
	*	2	F	IN...
	376		F	relation (1° du chemin)
	377		O	type de sélection
	378		F	relation (i° du chemin)
	379		F	paramètres de la sélection
	380		F	nom à redéfinir en mémoire utilisateur
	381		F	nom redéfini en mémoire utilisateur

<u>REGLES</u>	<u>N°</u>	<u>ATTRIBUT</u>	
	* 1	* 2	fait le lien avec la phrase MEMBER, précise le sous-schéma pour lequel la phrase sera utilisée
	376-378		si la sélection nécessite le parcours d'un chemin de relation
	380-381		si un nom appartient à plusieurs members d'l relation (set), il convient de pouvoir distinguer de quelle relation il s'agit

SIGN

FOR {element}
{group} -name OF version-number [IN SET set-name]

IS {LEADING }
{TRAILING} [SEPARATE CHARACTER]

DEFINITION

spécifie la position et le mode de représentation du signe opérationnel

DECRIE DANS
POUR

ENTITY

OCCURRENCE

0 - N

ELEMENT dans un SOUS-SCHEMA

"

0 - 1

TYPE

FACULTATIVE

DECOMPOSITION

N°	ATTRIBUT	TYPE	SIGNIFICATION
* 1		0	FOR nom element
* 2		0	IN nom sous-schéma
382		0	place du sign
383		F	separate character ?

REGLE

N° ATTRIBUT

383

si oui, le signe n'est pas 1 position
digit et la lettre S est comptée dans
la longueur

SOURCE

FOR element-name OF version-number IS $\left\{ \begin{array}{c} \text{ACTUAL} \\ \text{VIRTUAL} \end{array} \right\}$ OF

$\left\{ \begin{array}{c} \text{element} \\ \text{group} \end{array} \right\}$ -name OF OWNER OF relation-name

DEFINITION spécifie l'origine du contenu d'un élément et si cet élément est ou n'est pas physiquement contenu dans l'entité

<u>DECRIE DANS</u>	ENTITY	<u>OCCURRENCE</u>	0 - N
<u>POUR</u>	ELEMENT	"	0 - 1

TYPE FACULTATIVE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	*	0	FOR
	384	0	type de placement de l'item
	385	0	nom de l'élément source
	386	0	nom de relation

<u>REGLE</u>	<u>N° ATTRIBUT</u>	
	384	valeurs possibles : 1 , 2
	386	le record doit faire partie de cette relation

SUBSCHEMA

OF set-name [version-number]

DEFINITION nomme le schéma auquel ce sous-schéma appartient

<u>DECRIE DANS</u>	SET
<u>POUR</u>	SOUS-SCHEMA

OCCURRENCE 1

TYPE OBLIGATOIRE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	387	0	nom schéma

TEMPORARY

 {YES}
 {NO}

DEFINITION spécifie si le "set (area)" est temporaire ou pas

DECRIE DANS SET
POUR AREA

OCCURRENCE 1

TYPE OBLIGATOIRE

<u>DECOMPOSITION</u>	<u>N°</u>	<u>ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	388		0	temporaire ?

TYPEAPPART

MEMBER IS {MANDATORY} {AUTOMATIC} [LINKED TO OWNER]
 {OPTIONAL} {MANUAL}

[DUPLICATES NOT ALLOWED FOR {element}
 {group}-name [version-number]
 {element}
 {group}-name]

DEFINITION donne le type d'appartenance de l'entity à la relation et précise la non-duplication pour certains éléments de l'entity

DECRIE DANS RELATION OCCURRENCE 1 - N
POUR ENTITY (MEMBER) " 1

TYPE OBLIGATOIRE

<u>DECOMPOSITION</u>	<u>N°</u>	<u>ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	389		0	type d'appartenance (temps)
	390		0	" " (espace)
	391		F	linked to owner ?
	392		F	nom élément

<u>REGLE</u>	<u>N°</u>	<u>ATTRIBUT</u>	
	389		le record appartient-il toujours à la relation ?
	390		le record est-il toujours inséré dans la relation ?

TYPESET

IS (SCHEMA
SUB-SCHEMA
AREA
OTHER)

DEFINITION donne le type de set (le set regroupe divers aspects en CODASYL)

DECRIE DANS SET
POUR AREA, SOUS-SCHEMA, SCHEMA

OCCURRENCE 1 - N 1 - N 1

TYPE OBLIGATOIRE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	*	0	type de set

REGLE N° ATTRIBUT
 * valeurs possibles : 1 — 9
 OTHER pourvoit à l'éventualité d'une découpe logique

REMARQUE - cette phrase permet de représenter sous forme de set tous les concepts repris dans la liste des valeurs d'attribut; cette phrase est obligatoire, il convient de ne l'oublier en aucun cas

USAGE

FOR {element} -name OF version-number [IN SET set-name] IS (COMP
 {group}) (COMP-n
) (DISPLAY
) (DATABASE-KEY)

DEFINITION spécifie le format des elements COBOL et demande certaines conversions entre l'element tel qu'il est décrit dans le schéma et ce qu'on désire avoir dans la zone de l'utilisateur

<u>DECRIE DANS</u>	ENTITY	<u>OCCURRENCE</u>	
<u>POUR</u>	ELEMENT dans SOUS-SCHEMA	"	0 - N 0 - 1

TYPE FACULTATIVE

<u>DECOMPOSITION</u>	<u>N° ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	*	0	FOR...
	*	0	IN...
	393	0	type de représentation

REGLE N° ATTRIBUT
 393 valeurs possibles : 1, 2, 3, 4
 COMP-n le numéro dépend du type de machine

<u>USERNAME</u>

IN SCHEMA IS entity-name [version-number]DEFINITION

nomme l'entity maître de celle du sous-schéma (étant donné qu'en général on ne reprend pas toutes les composantes de l'entity du schéma dans le sous-schéma)

DECRIE DANS
POUR

ENTITY

ENTITY organique dans sous-schéma

OCCURRENCE

1

TYPE

OBLIGATOIRE

DECOMPOSITION

<u>N°</u>	<u>ATTRIBUT</u>	<u>TYPE</u>	<u>SIGNIFICATION</u>
	*		nom de l'entity dans schéma

REMARQUE

l'analyseur remplacera le nom de l'entity : du sous-schéma
par le USERNAME

LISTE DES ATTRIBUTS DEDUITS (de phrases du PSL)

nom de tout objet			
version de tout objet			
responsable-problem-definer	420	SET	(clause même nom)
responsable-interface	421	SET	(clause même nom)
french-description	422	ENTITY	(clause même nom)
nether-description	423	ENTITY	(clause même nom)
objets dans CONTAINED IN	424	ENTITY	
objets dans CONSISTS OF	425	ENTITY	
nom de l'owner	426	RELATION	(between-and-)
nom du member	428	RELATION	(between-and-)
n° de niveau	431	à déduire de la structure	
facteur de répétition	432	ENTITY-GROUP (consists of <u>n</u>)	
type de groupe-élément	410-411-412	à déduire à partir des clauses INTERNAL-FORM, TYPE,...	

REMARQUES: - on reprend des informations du PSL qu'on désire
 utiliser pour décrire une base de données.
 - ces informations sont aussi des attributs.

DESCRIPTION DES SQUELETTES.

Chaque squelette décrit une partie d'imprimé. Il est fait de textes, de conditions et de paramètres. Chacune de ces informations est définie et son origine est précisée.

Légende



si la condition A est vraie, elle implique la présence du paramètre B



si la condition A est vraie, elle implique l'impression de texte



si la condition A est vraie, un traitement B doit s'effectuer
si la condition A est fausse, un traitement C doit s'effectuer

n° attribut

n° attribut d'où proviennent les valeurs de ce paramètre et/ou condition

NO

pointeur vers les valeurs de cet attribut est garni

NB














nombre de valeurs de cet attribut est supérieur à l'unité

Remarque

- pour constater la part d'imprimé que couvre chaque squelette, l'utilisateur est prié de consulter le "SAMPLE OUTPUT" et de suivre sur la colonne d'extrême gauche

SCHEMA
DDL

CONDITION =====	PARAMETRE =====	POSITION =====	N° ATTRIBUT =====	SIGNIFICATION =====
1 SCHDDL	NAME			nom du "schéma"
	RESPRO		420	nom du responsable
	RESINT		421	nom de l'interface
	VERS			
	LITT		318(NO) 318(NB)	littéral de PRIVACY d'autres littéraux?
	PRIVAC → 1 OTHERP → 0			
2 PRVDDL	LITT		318 318(NB)	
	OTHERP → 0			
3 TARDDL				titre de l'area sec- tion'
4 AREDDL	NAME			nom de l'area"
	RESPRO		420	
	RESINT		421	
	VERS			
	TEMPOR → 1		388(NO)	"area" est temporaire
5 TREDDL				titre de la "record section"
6 ENTDDL	NAME			nom de l'entité
	VERS			
	FRDES1		422	description en langue française
	FRDES2		422	
	FRDES3		422	
	FRDES4		422	
	NTDES1		423	beschrijving in nederlandse taal
	NTDES2		423	
	NTDES3		423	
	NTDES4		423	
	KEYLOC		330	clé du "location MODE DIRECT"
	LOCDIR → 1			
	LOCCAL → 1		328	procédure du "LOCATION MODE CALC"
	KEYLOC		331	clé du "LOCATION MODE CALC"

7 LOCDDL	KEYLOC	331	clé de "LOCATION MODE CALC"
8 DUPDDL	DUPLNO 	332	"duplicates are not allowed"?
9 WITDDL	IMP → END1  END2 	NAME NEWNM 424(NB)	impression de "WITHIN" ? nom d'une "area" impression de "AREA ID IS.."?
10 SEAENT	IMP → NAME	371	impression de "SEARCH KEY IS"? nom de clé de "SEARCH"
11 USIENT	TYPIDX  TYP1  INDEX 	PROCNM INDXNM 372 373 374	y-a-t'il un index ? procédure du CALC ? nom de l'index ?
12 DUPSER	DUPLNO  END1 	375	"duplicates are not allowed"?
13 TEGDDL	NAME		nom de l'entité ("record")
14 ELGDDL	NIVEAU  PICTUR  TYPE  INTEGR  OCCRS 	NAME NONIV INFPIC TYPCAR ENTIER ENTR2 FACTRP POSIT1 POSIT2 POSIT2 411 POSIT3 POSIT2 425 431 355 410 411 412 432	nom d'un élément ou groupe n° de niveau ? "PICTURE" ? "TYPE" ? autre entier de "TYPE" ? "OCCURS" ?
15 TSEDDL			titre de relation ("set")

16

SETDDL

	NAME			nom de relation ("set")
	OWNER	426		
	MEMBER			"set is dynamic" ?
DYNAM → 1				
	ORDER	349		type d"ORDER"
INDEX → 1		351		index présent ?
NAMED → 1	INDXNM	352		index nommé ?
SORTED → 1	TYPsor	350		type de "sorted"
DUPL → 1		354		clause duplicates présente ?
DUPLNO → 1		354		"duplicates are not allowed" ?

17

MEMDDL

	NAME	333		nom d'un "membre"
	TYPAP1	389		"mandatory" "optional" ?
	TYPAP2	390		"automatic" "manual"
END → 1	.			

18

KEYDDL

IMP →	NAME	325		impression de "KEY IS" ?
	TYPKEY	323		nom de clé de la clause "KEY"
END1 → 1	.			type de clé de la clause "KEY"

19

SELDDL

TYP1 → 1		378		type de "SET OCC. SELECTION"
END1 → 1	.			

20

ALIDDL

	NAME	380		nom à renommer
	NEWNM	381		nom renommant
END1 → 1	.			

SCHEMA
SSL

CONDITION =====	PARAMETRE =====	POSITION =====	N° ATTRIBUT =====	SIGNIFICATION =====
21 SCHSSL	NAME			nom du "schéma"
22 ARESSL	NAME			nom d'une "area"
23 TRESSL				
24 RECSSL	NAME			nom d'entité ("record")
DBKEYT →	ENTIER		313-314(NO)	clause "DBKEY-TRI-TABLE"
INTEGR →			313	entier de clause "DBKEY-TRI-TABLE"
REAL →	AREANM		314	area de clause "DBKEY-TRI-TABLE"
POPUL →	ENTR2		357 (NO)	clause "POPULATION" ?
END1 →	REALMN		359	
	.			
25 PO1SSL	ENTR2		357	
END1 →	REALMN		359	
	.			
26 PLASSL	NAME		356	nom de l'area de "PLACEMENT"
END1 →	.			
27 IXRSSL	INDXNM		319	nom de l'index
PLACEM →	REALMN		320	nom de l'area de l'index
TYP 1 →			322	type de l'index
TYPIDX →			322	type de l'index
REORG →	ENTIER		362-361(NO)	clause "REORGANIZATION"
END1 →	.			
28 COMSSL				

titre de relation SSL

entête de relation SSL

type de mode
"linked" ?

clause "POPULATION"

clause "REORGANIZATION"

placement de l'index

type de l'index

clause "REORGANIZATION"

"member is physically
linked to owner"

titre fin de schéma

NAME

MODE

REALMN

ENTIER

ENTR2

ENTIER

REALMN

ENTIER

334

335

337

337

337

338

357

358

361

320

320

321

322

322

361

LINKED → 1

MODE1 → 1

MODE2 → 1

MODE3 → 1

WITHIN → 1

INTEGR → 1

END1 → 1

END1 → 1

PLACEM → 1

PLACM2 → 1

WITHIN → 1

TYP1 → 1

TYPIDX → 1

REORG → 1

END1 → 1

29

TSWSSL

30

SETSSL

31

MODSSL

32

PO2SSL

33

REOSSL

34

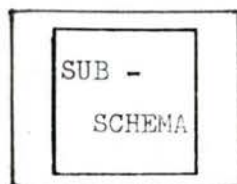
IXSSSL

35

MBPSSL

36

ENDSCH



CONDITION	PARAMETRE	POSITION	N° ATTRIBUT	SIGNIFICATION
	NAME			nom du "sous-schéma"
	SCHENM		387	nom du "schéma"
	RESPRO		420	nom du responsable
	RESINT		421	nom de l'interface
	VERS			
PRIVAC → 1	LITT		318 (NO)	clause "PRIVACY" ?
END1 → 1	.			
	LITT		360	clause "PRIVACY KEY"
IMP →				impression du titre
				"area" ?
END1 → 1	REALMN			nom d'une "area"
				titre de l'entity
				section "
	NAME			entête d'une entité
				("record")
NIVEAU → 1	NONIV	POSIT1	431	n° de niveau ?
	NAME	POSIT2	425	
PICTUR → 1	INFPIC	POSIT2	355	"PICTURE" ?
USAGE → 1	INFUSG	POSIT2	393	"USAGE" ?
OCCRS → 1	FACTRP	POSIT2	339	"OCCURS" ?
IMP → 1	NAME		306	impression du nom de
				la condition
END2 → 1			306 (NB)	
PLUSLT → 1	LITT		308	littéral début valeurs
OTHERL → 1	LITT2		309	littéral fin valeurs
END1 → 1	.			

44

SETSUB

IMP →

END1 →

NAME

.

impression de la "set
section" ?
nom de relation

45

ENDSUB

fin des sous-schémas

E R R O R

46

ERREUR

NAME

ERR1

ERR2

nom de l'objet fautif
libellé de l'erreur
(partie 1)
libellé de l'erreur
(partie 2)

Sortie Générée.

La sortie générée représente l'imprimé exact maximal à générer.
L'information sous cadre est numérotée.
Ce numéro renvoie à une liste où pour chaque information sont
renseignés son origine et ses attributs concernés.

Legende.

- | | | |
|----|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 1. | <div style="border: 1px solid black; width: 150px; height: 1.2em; display: inline-block;"></div> | ensemble de 12 caractères |
| 2. | <div style="border: 1px solid black; width: 150px; height: 1.2em; display: inline-block; text-align: center;">?</div> | ensemble de caractères de longueur indéterminée |
| 3. | <div style="border: 1px solid black; width: 150px; height: 1.2em; display: inline-block; text-align: center;">a</div> | répétition du même type d'ensemble de caractères sur les lignes suivantes |

Tout ce qui n'est pas sous cadre est créé par le générateur.
Le numéro présent à gauche est celui du squelette réalisant
cette partie d'imprimé.
Ce numéro ne sera pas imprimé.

```

+ + + + + SCHEMA DESCRIPTION STATEMENTS
+
+ RESPONSIBLE-INTERFACE IS [A]
+ RESPONSIBLE-DEFINER IS [B]
+ VERSION NUMBER IS [C]
+
+ SCHEMA NAME IS [D]
+ PRIVACY LOCK FOR COPY IS [E]
+ AREA DESCRIPTION STATEMENTS
+
+ RESPONSIBLE-INTERFACE IS [F]
+ RESPONSIBLE-DEFINER IS [G]
+ VERSION NUMBER IS [H]
+
+ AREA NAME IS [I]
+ [J] AREA IS TEMPORARY
+ RECORD DESCRIPTION STATEMENTS
+
+ VERSION NUMBER IS [K]
+ FRENCH-DESCRIPTION
+ NETHER-DESCRIPTION
+
+ [L] ? v
+ [M] ? v

```

```

RECORD NAME IS
7-8 LOCATION MODE IS
9 WITHIN
AREA-ID IS
10-11-12 SEARCH KEY IS
13 ++++++
+ ITEMS OR GROUPS IN
+ ++++++
+ ++++++
14 [SI] [T1]
PICTURE IS
TYPE IS
OCCURS TIMES.

```

```

+ ++++++
15 + SET DESCRIPTION STATEMENTS
+ ++++++
+ ++++++
+ ++++++
+ SET NAME IS
16 + OWNER NAME IS
+ MEMBER NAME
+

```

```

SET NAME IS
[A2 SET IS DYNAMIC]
ORDER IS
OWNER IS
17 MEMBER IS
18 [D2] KEY IS
10-11-12 SEARCH KEY IS
19-20 + SET OCCURRENCE SELECTION
+ 7 8 12

```


AI	Ordre "RESPONSIBLE-INTERFACE" (attribut 421)	SET (TYPESET=SCHEMA)
BI	élément "RESPONSIBLE-PROBLEM-DEFINER" (attribut 420)	SET (TYPESET=SCHEMA)
CI	n° de version	SET (TYPESET=SCHEMA)
DI	nom (paragraphe définition)	SET (TYPESET=SCHEMA)
EI	phrase "DBPRIVACY" (attribut 318)	SET (TYPESET=SCHEMA)
FI	ordre "RESPONSIBLE-INTERFACE" (attribut 421)	SET (TYPESET=SCHEMA)
GI	élément "RESPONSIBLE-PROBLEM-DEFINER" (attribut 420)	SET (TYPESET=AREA)
HI	n° de version	SET (TYPESET=AREA)
II	nom (paragraphe définition)	SET (TYPESET=AREA)
JI	phrase "TEMPORARY" (attribut 388)	SET (TYPESET=AREA)
KI	n° de version	SET (TYPESET=AREA)
LI	clause "DESCRIPTION" (attribut 422)	ENTITY
MI	clause "DESCRIPTION" (attribut 423)	ENTITY
NI	nom (paragraphe définition)	ENTITY
OI	phrase "LOCATION" (attributs 327, 328, 330, 331, 332)	ENTITY
PI	ordre "CONTAINED IN" (on reprend les sets où TYPESET=AREA) (attribut 424)	ENTITY
QI	créé par le générateur : ARAD + nom entity NI	ENTITY
RI	phrase "SEARCH" (attributs 371, 372, 373, 374, 375)	ELEMENT-GROUP
SI	n° de niveau (attribut 431)	ELEMENT-GROUP
TI	nom (paragraphe définition)	"
SI-TI	attribut 425 on donne la liste (CONSISTS OF)	"
UI	attribut 355	"
VI	attributs 410, 411, 412	GROUP-ENTITY
WI	ordre "CONSISTS OF" du niveau immédiatement supérieur (attribut 432)	RELATION
XI	nom (paragraphe définition)	RELATION
YI	clause "BETWEEN" (nom entity 1) -- (attribut 426)	RELATION
ZI	clauses "MEMBER" ou <u>entity 2</u> ... de la clause BETWEEN -- (attribut 428)	RELATION
A2	si OWNER IS SYSTEM	RELATION
B2	phrase "ORDER" (attributs 349, 350, 351, 352, 354)	RELATION
C2	phrase "TYPEAPPART" (attributs 389, 390)	RELATION
D2	phrase "KEY" (attribut 323)	RELATION
E2	phrase "KEY" (attribut 325)	RELATION
F2	phrase "SEARCH" (attributs 371, 372, 373, 374, 375)	RELATION
G2	phrase "SELECTION" (attributs 377, 380, 381)	RELATION

H2	créé par le générateur : DCB + nom du set area HI	ENTITY
I2	phrase "DBKTRLTB" (attributs 313, 314)	ENTITY
J2	phrase "POPULATION" (attributs 357, 359)	ENTITY
K2	phrase "PLACEMENT" (attribut 356)	ENTITY
L2	phrase "INDEX" (attributs 319, 320, 321, 322, 362)	ENTITY
M2	phrase "COMPRESSION" (attribut 305)	ENTITY
N2	phrase "MODE" (attributs 334, 335, 337, 338)	RELATION
O2	phrase "POPULATION" (attributs 357, 358)	RELATION
P2	phrase "REORGANIZATION" (attribut 361)	RELATION
Q2	phrase "INDEX" (attributs 319, 320, 321, 322, 362)	RELATION
R2	si linked to owner	RELATION

37 IDENTIFICATION DIVISION.

+
+
+ SUB-SCHEMA DESCRIPTION STATEMENTS
+
+
+ RESPONSIBLE-INTERFACE IS T2
+ RESPONSIBLE-DEFINER IS U2
+ VERSION NUMBER IS V2
+

SUB-SCHEMA NAME IS W2
OF SCHEMA NAME X2

2 PRIVACY LOCK FOR COMPILE IS Y2
38 PRIVACY KEY FOR COPY IS Z2
39 DATA DIVISION.

AREA SECTION.

+
+
+ AREA DESCRIPTION STATEMENTS
+
+
+

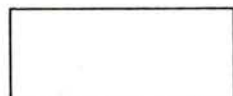
COPY

A3

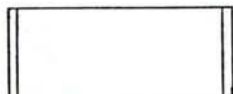
S U B - S C H E M A

T2	ordre "RESPONSIBLE-INTERFACE" (attribut 421)	SET (TYPESET=S-SCHEMA)
U2	ordre "RESPONSIBLE-PROBLEM-DEFINER" (attribut 420)	SET (TYPESET=S-SCHEMA)
V2	n° de version	SET (TYPESET=S-SCHEMA)
W2	nom (paragraphe définition)	SET (TYPESET=S-SCHEMA)
X2	attribut "SUBSCHEMA"	SET (TYPESET=S-SCHEMA)
Y2	phrase "DBPRIVACY" (attribut 318)	SET (TYPESET=S-SCHEMA)
Z2	phrase "PRIVACY KEY" (attribut 360)	SET (TYPESET=S-SCHEMA)
A3	sous-schéma <u>consists of</u> entity <u>contained in</u> set (area)	SET (TYPESET=S-SCHEMA)
B3	noms des entités reprises dans la clause "CONSISTS OF"	SET (TYPESET=S-SCHEMA)
C3	idem S1	
D3	idem T1	
E3	phrase "PICTURE" (attribut 355)	ELEMENT - GROUP
F3	phrase "USAGE" (attribut 393)	ENTITY
G3	phrase "OCCURS" (attribut 339)	ENTITY
H3	" " "CONDITION" (attributs 306, 308, 309)	ENTITY
I3	sous-schéma <u>subparts are</u> areas <u>consists of</u> entity <u>related to</u> RELATION	SET (TYPESET=S-SCHEMA)

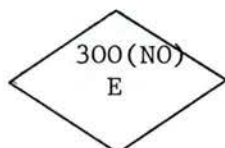
Organigrammes des programmes PRDBGDDL, PRDBGSSL, PRDGSUB

Légende


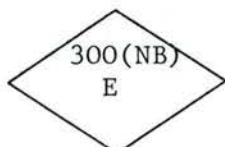
Représente une opération logique indécomposable



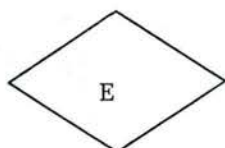
Représente une opération logique décomposable (son organigramme est précisé)



Si le pointeur vers la valeur de cet attribut est différent de zéro, procéder au traitement de cet attribut.



Si le nombre de valeurs de cet attribut est supérieur à l'unité, itérer le traitement de cet attribut jusqu'à la fin de sa liste de valeurs.



Dans l'index des entités (S=SET, E=ENTITY, R=RELATION, GE=GROUP-ELEMENT)

T

Traitement

Q

Question

T-XX

Signifie que l'opération logique consiste à effectuer le traitement de XX c'est-à-dire, dans la majeure partie des cas :

1. lire le pointeur vers la valeur de l'attribut correspondant à XX
2. lire la table de valeur de l'attribut correspondant à XX
3. placer cette valeur dans la table pour PRGENSKE à l'adresse correspondante à XX

Toutefois, le nom de paragraphe peut parfois

1. signifier une découpe logique en vue de clarifier l'organigramme
2. masquer une autre opération logique additionnée d'un traitement se limitant à 2 voire 4 instruction.

exemple : 1. FIRST-PART-ENTITY, T-ENTITY, ...

2. T-SEARCH-R-NAME → T-S-NAME, ...

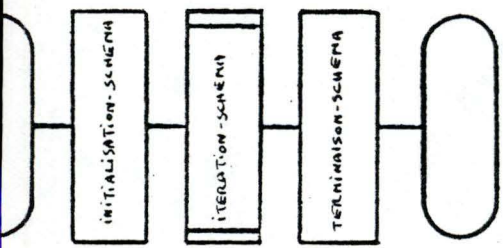
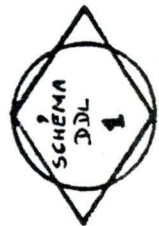
Q-XX

Signifie que l'opération logique T-XX est à effectuer si le pointeur vers la valeur de l'attribut correspondant à XX est garni.

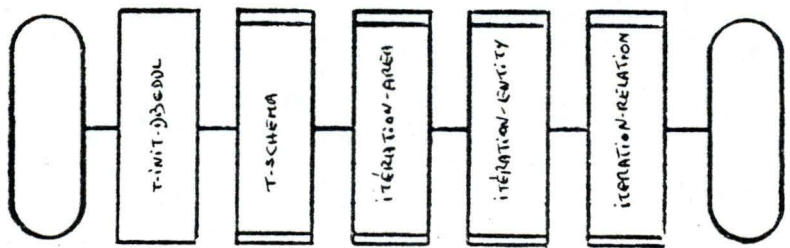
Toutefois, le nom de paragraphe peut parfois servir à définir des tests de fin de clauses ou d'objets.

- T-SKEL-i Réalise l'appel au générateur de sortie (PRGENSKE) en lui donnant
1. les tables de paramètres, conditions et positions.
 2. le numéro (=i) du squelette à traiter.
- T-XX-COPY XX fait partie des modules communs aux 3 programmes et présents dans la section qui est COPY

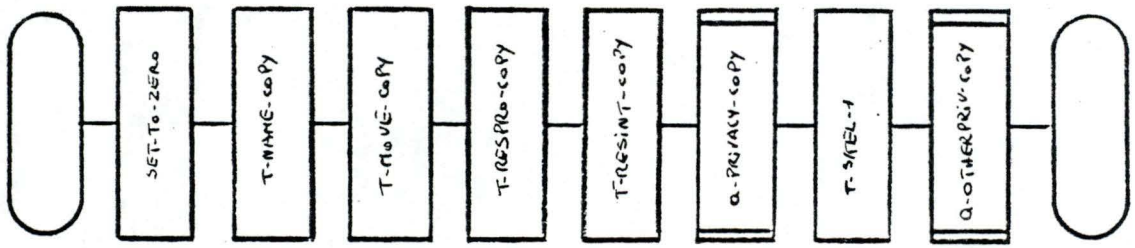
REMARQUE: le lecteur est invité à consulter en parallèle les annexes A5.1 et A5.2 qui reprennent l'ensemble des clauses du schéma et du sous-schéma UDS/V2



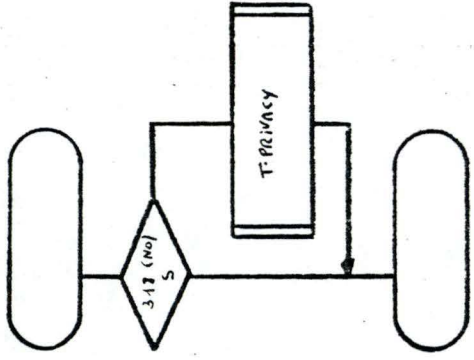
Root Of The Diagram



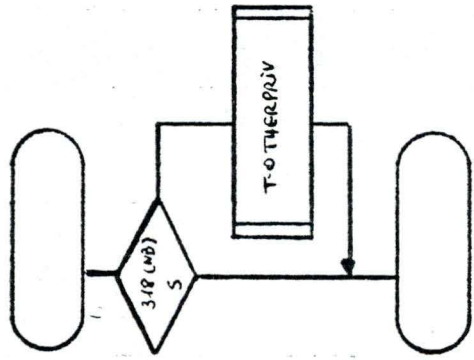
ITERATION-SCHEMA



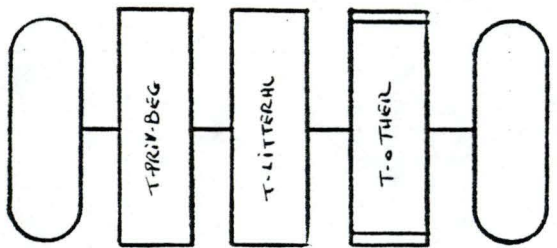
T-SCHEMA



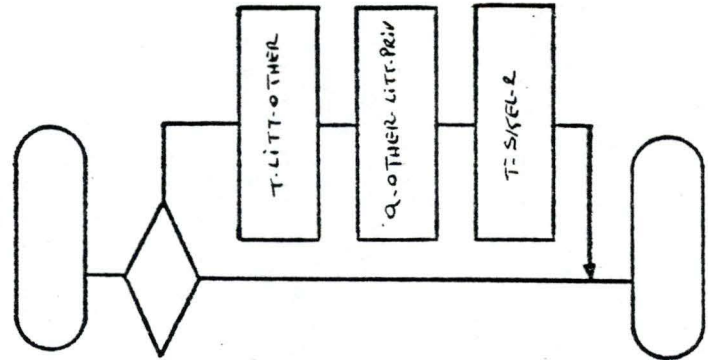
Q-PRIVACY-COPY



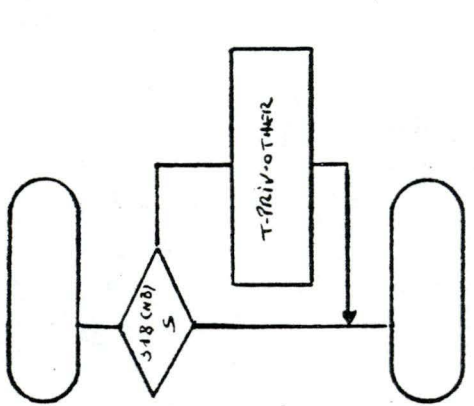
Q-OTHERPRIV-COPY



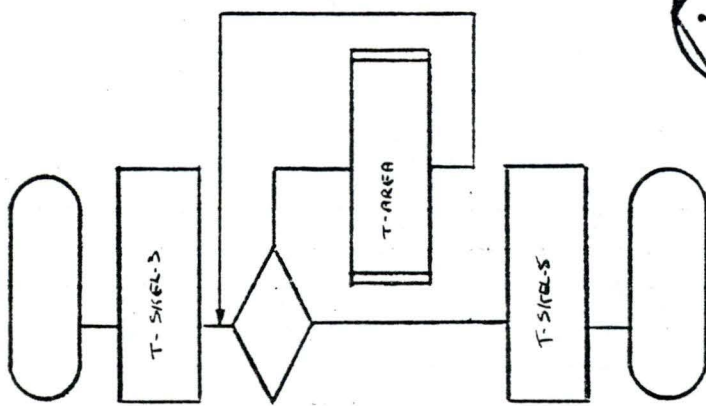
T-PRIVACY



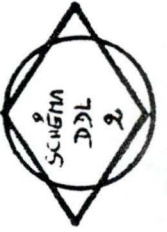
T-OTHERPRIV



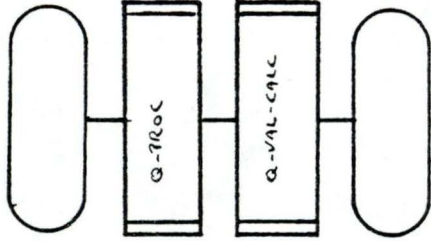
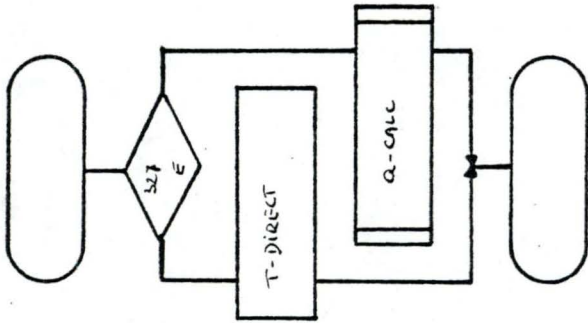
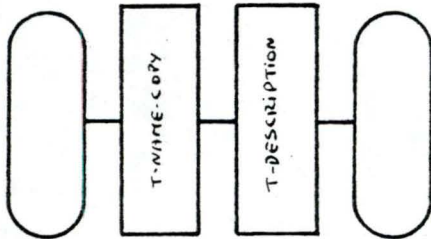
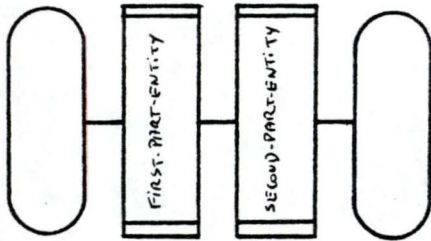
T-OTHER



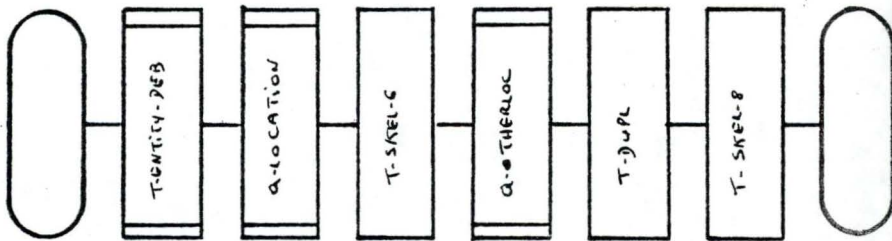
ITERATION-AREA



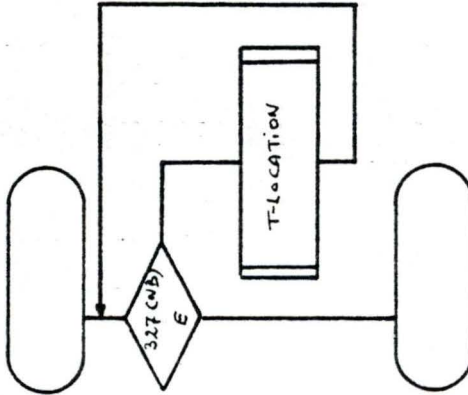
T-ENTITY



FIRST-PART-ENTITY

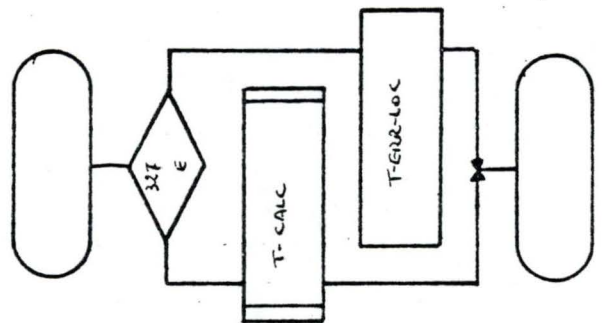


T-ENTITY-DEB



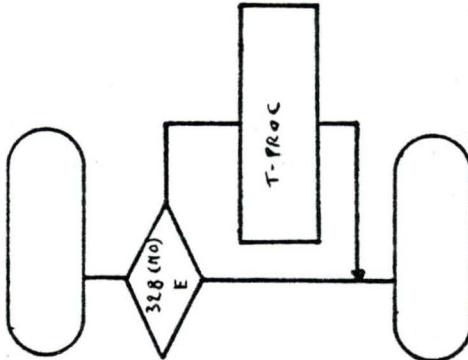
Q-LOCATION

T-LOCATION



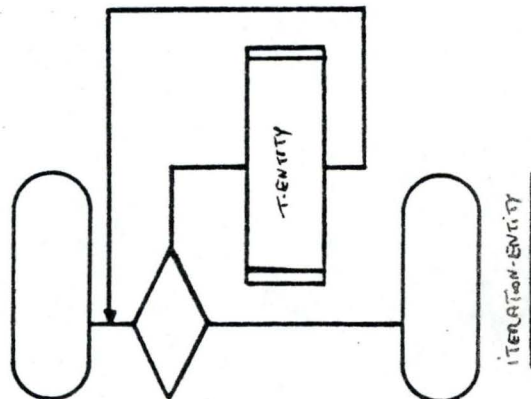
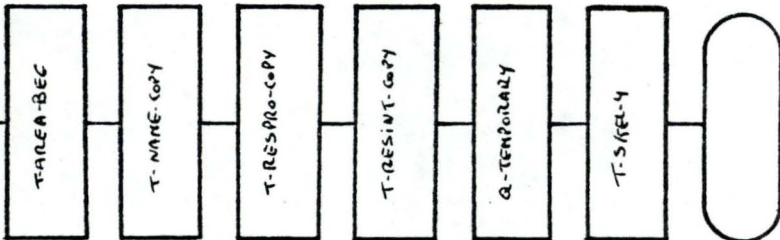
Q-CALC

Q-PROC

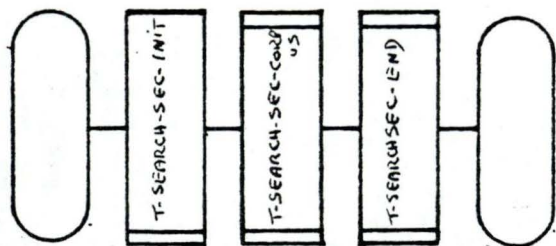
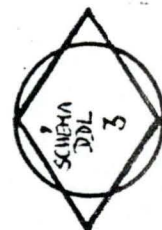


T-CALC

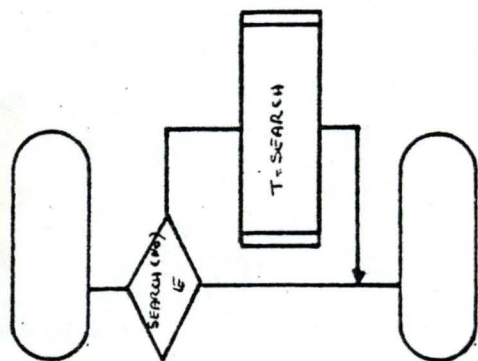
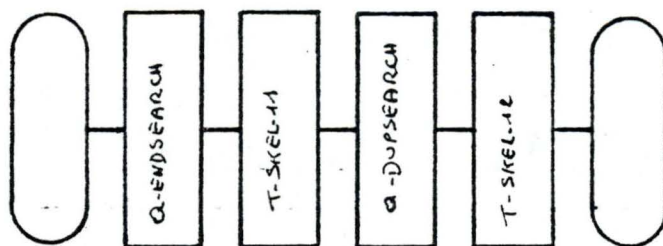
T-AREA



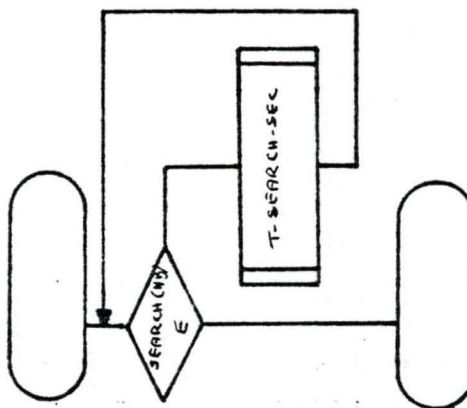
ITERATION-ENTITY



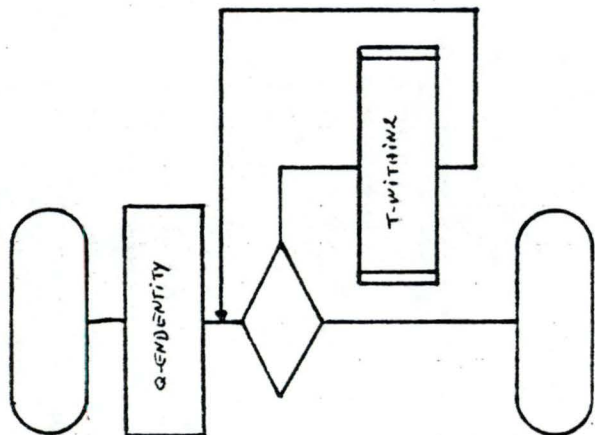
T-SEARCH-SEC



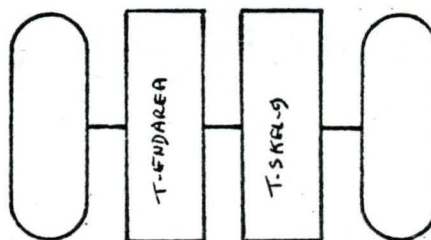
Q-SEARCH



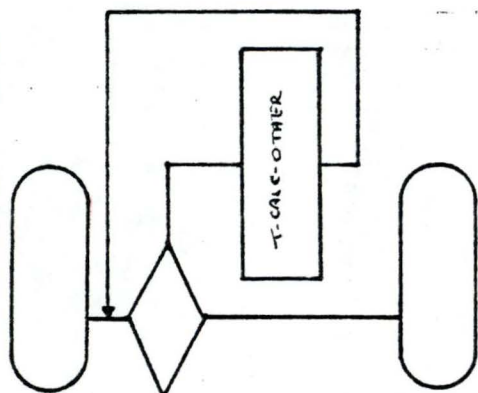
T-SEARCH



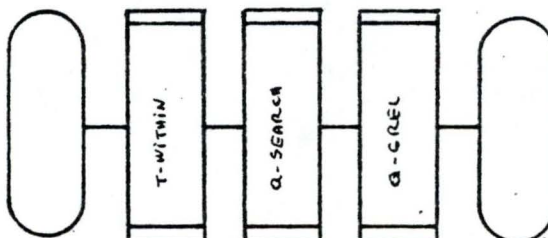
T-WITHIN



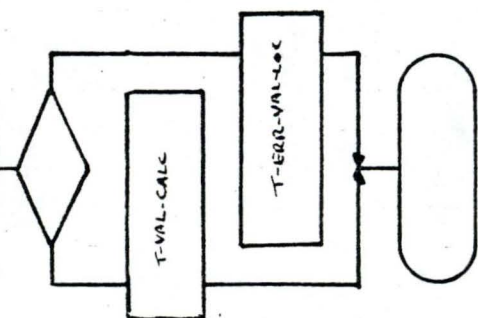
T-WITHIN2



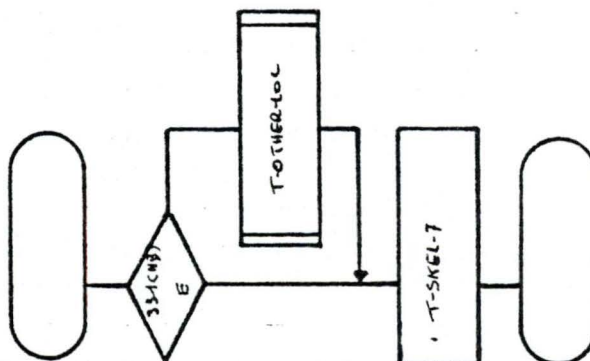
T-OTHER-LOC



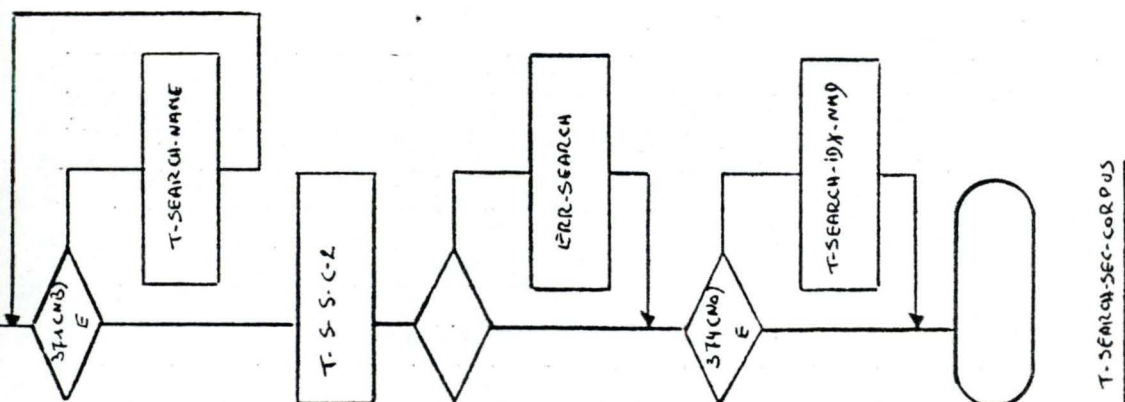
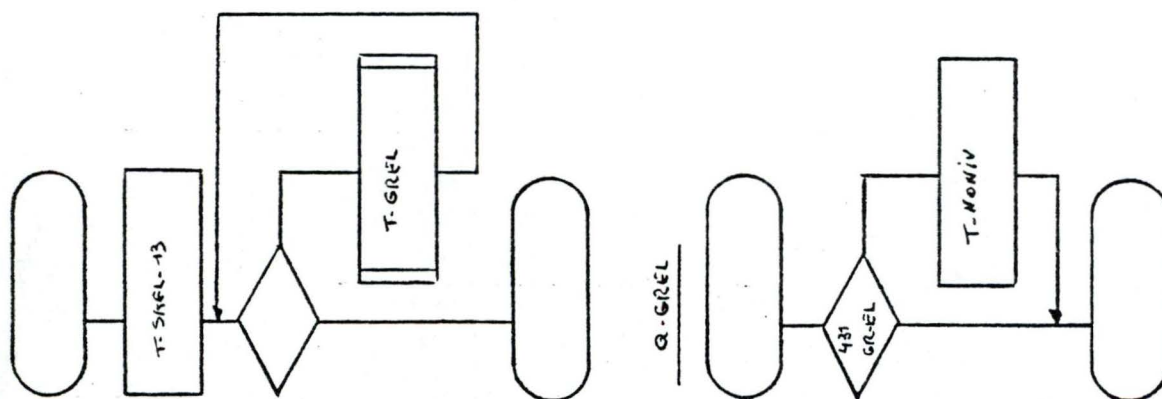
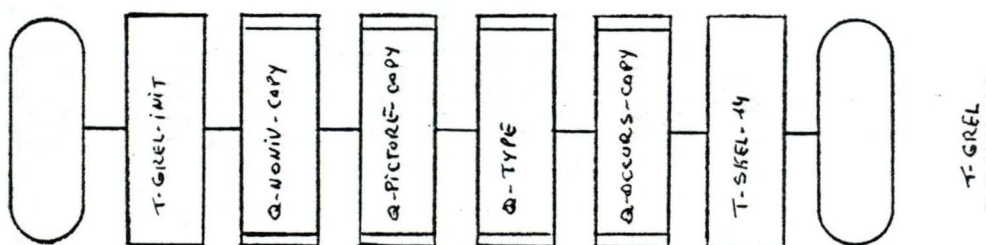
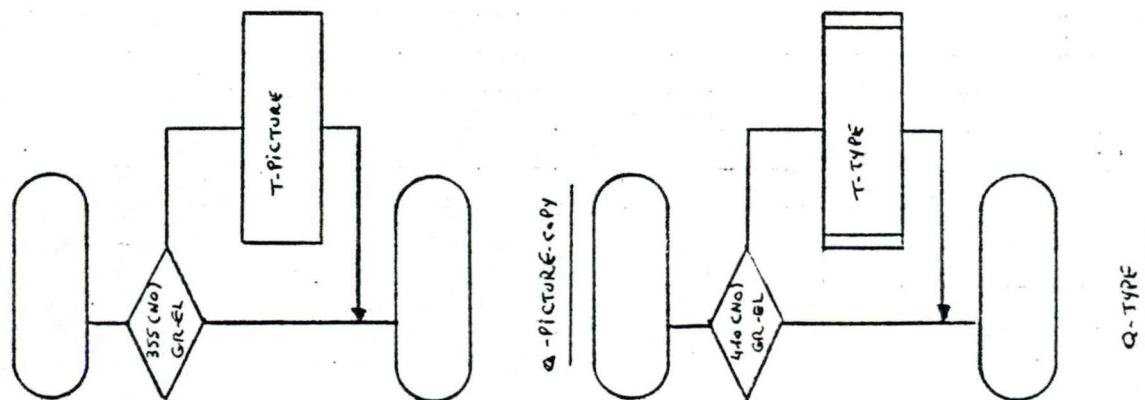
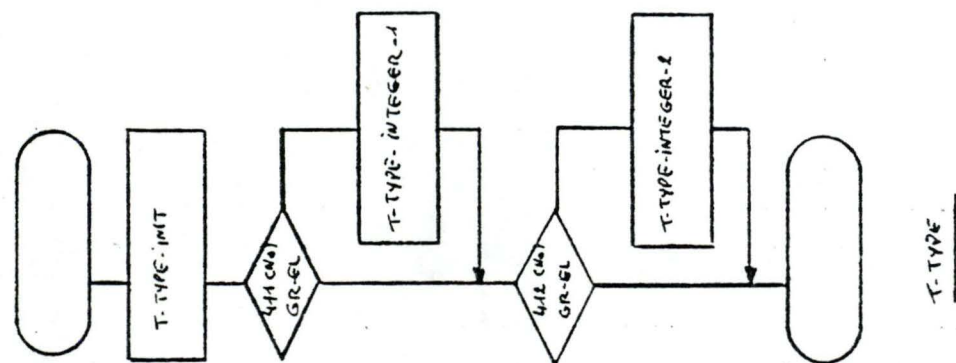
SECOND-PART-ENTITY



Q-VAL-CALC

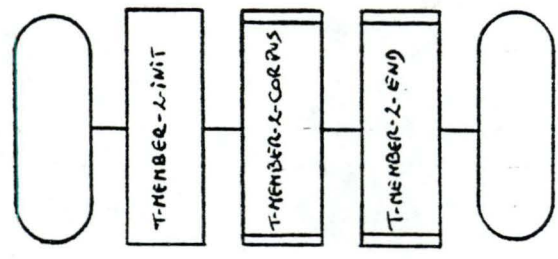


Q-OTHER-LOC

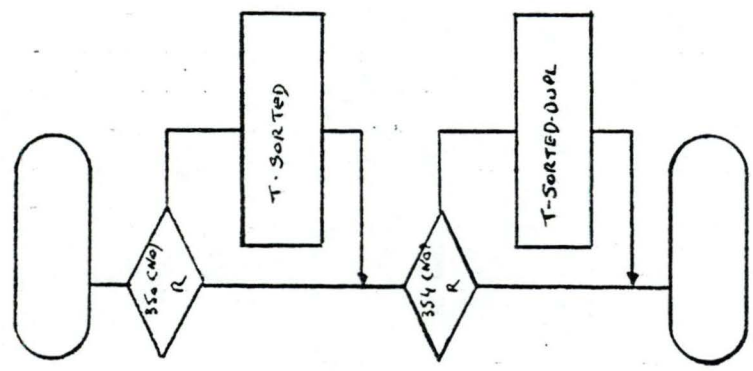
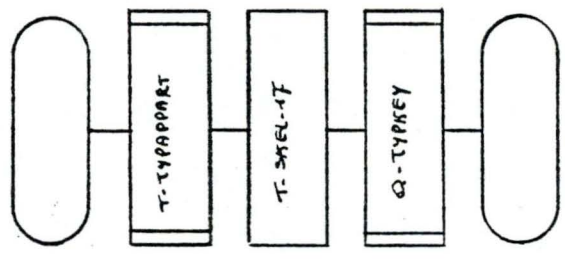




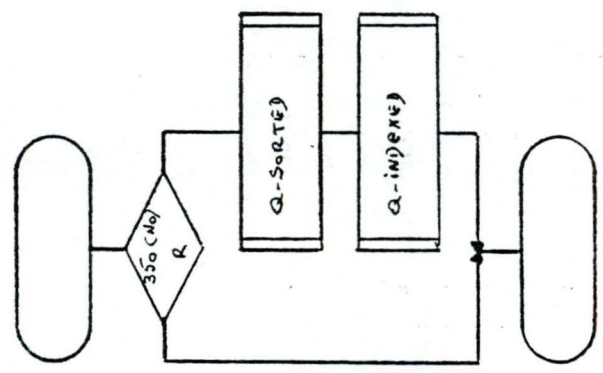
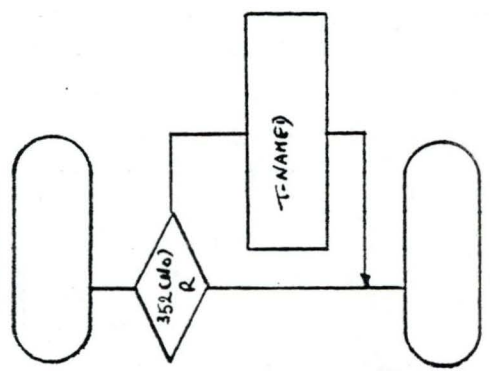
T-MEMBER-2-CORP
US



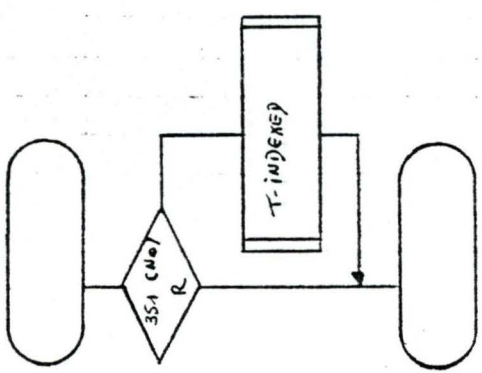
T-MEMBER-2



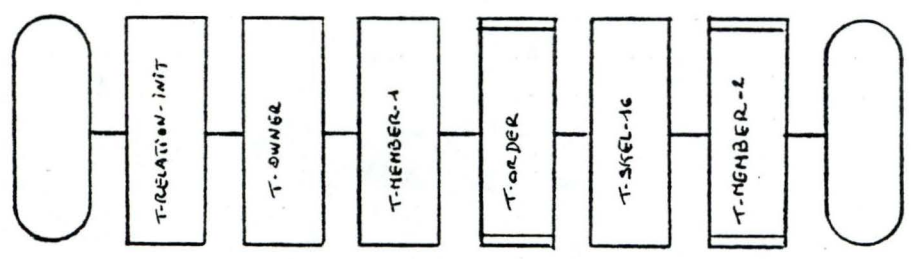
Q-SORTED



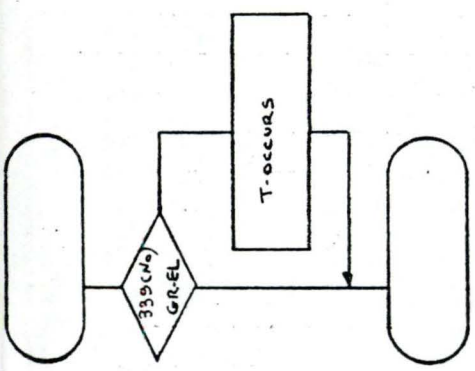
T-ORDER



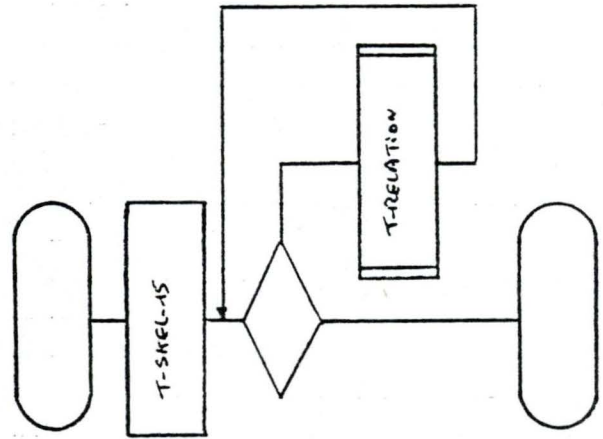
Q-INDEXED



T-RELATION



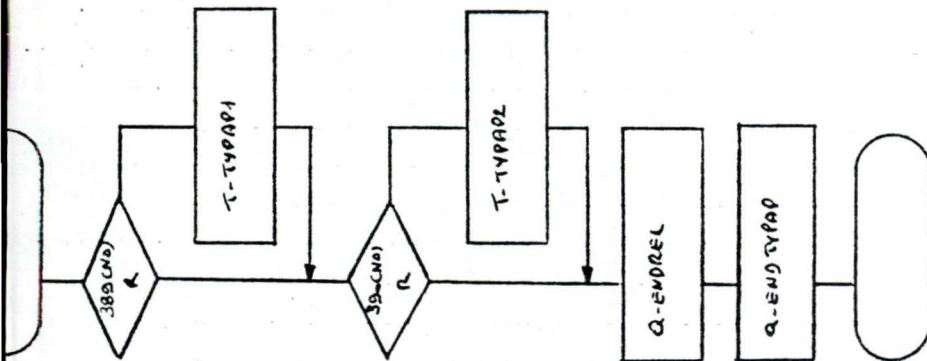
Q-occurs



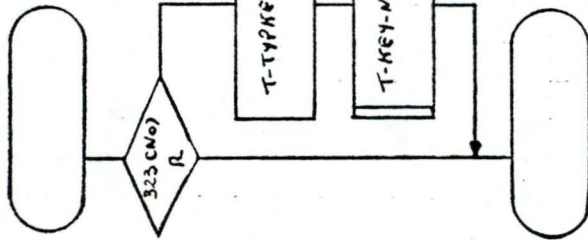
Iteration-Relation



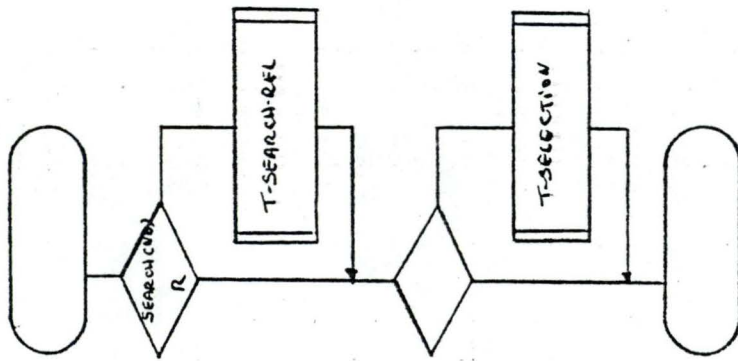
a. TYPEKEY



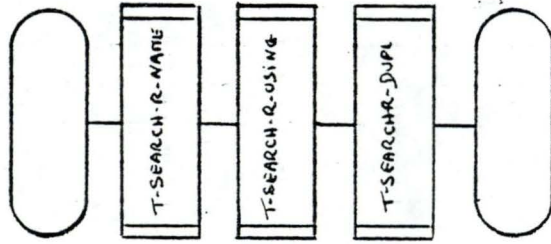
T-TYPAPART



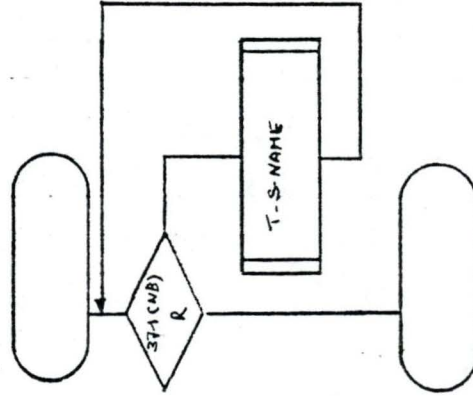
T-KEY-NAME



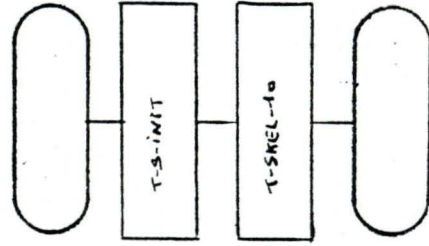
T-MEMBER-2-END



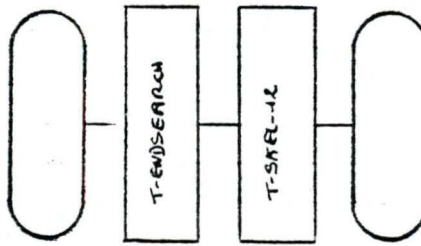
T-SEARCH-UNIT



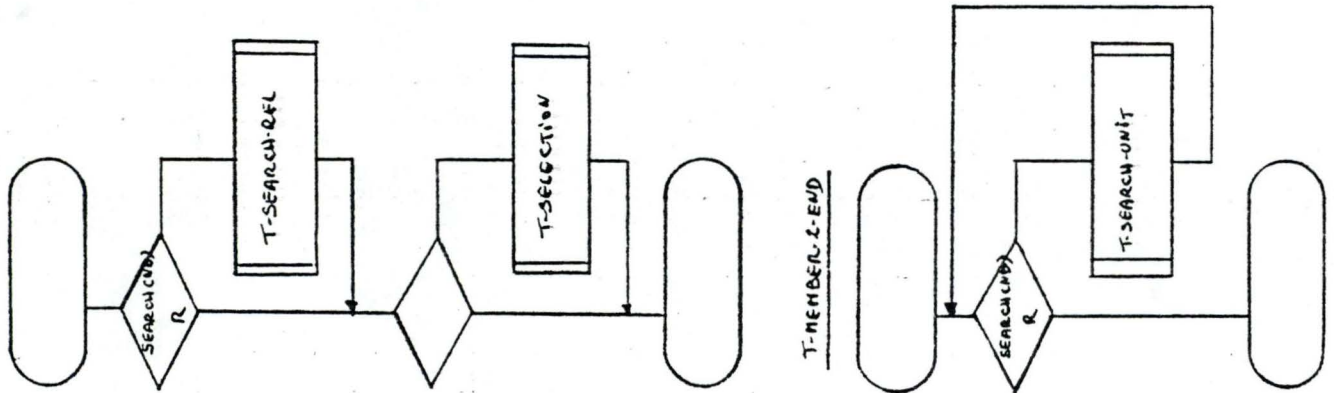
T-SEARCH-R-NAME



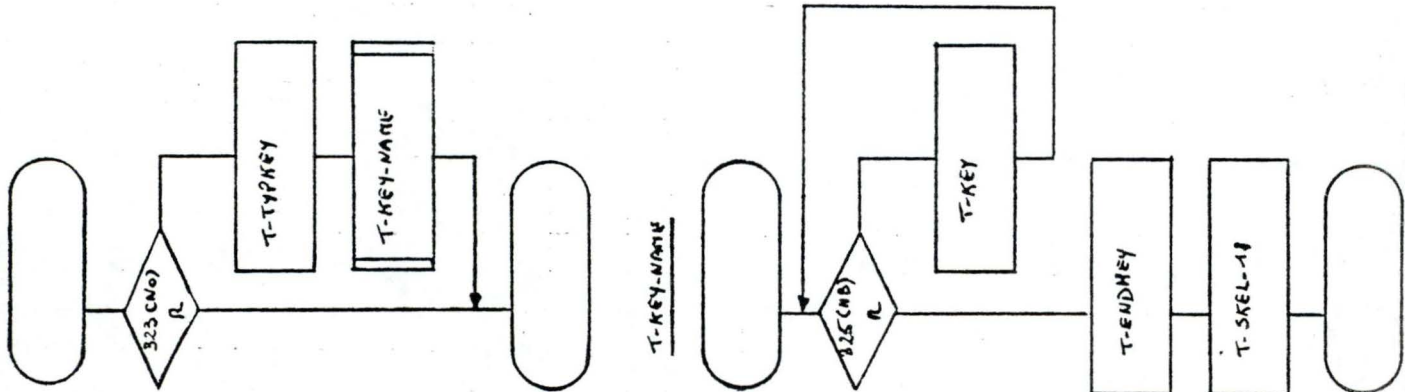
T-S-NAME

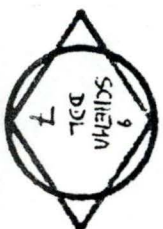
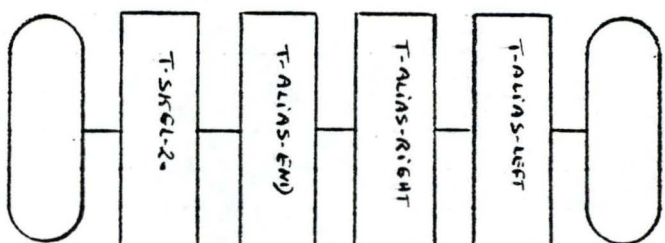
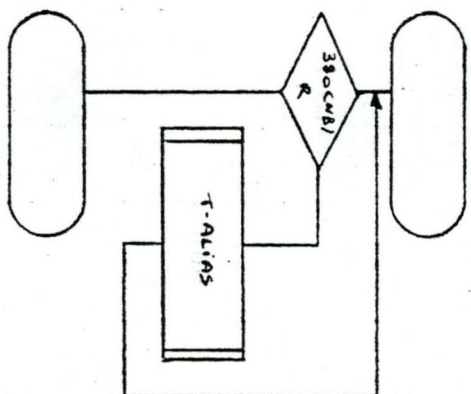
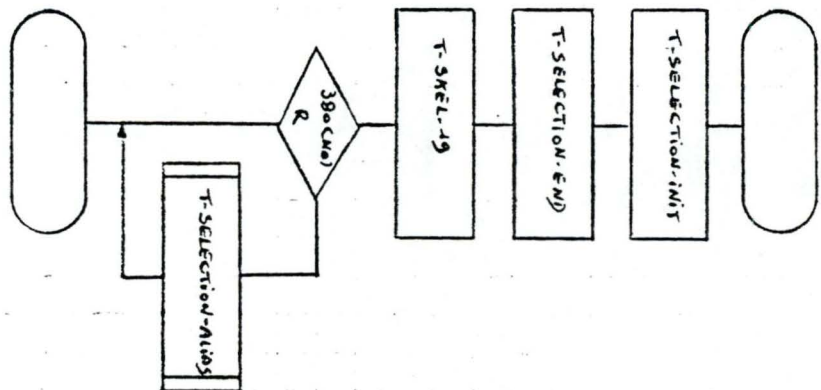
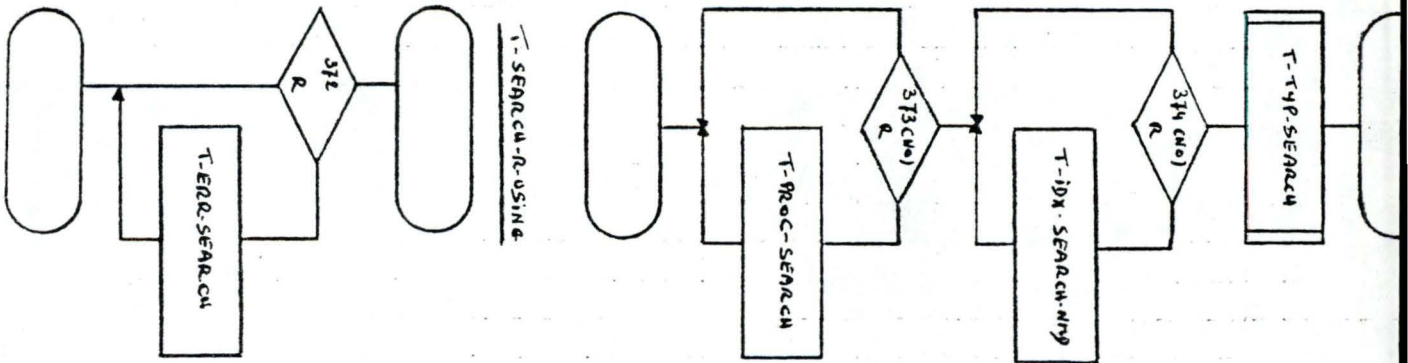


T-SEARCH-R-DUPL



T-SEARCH-REL





SCHEMA DDL (PRDBGDDL)

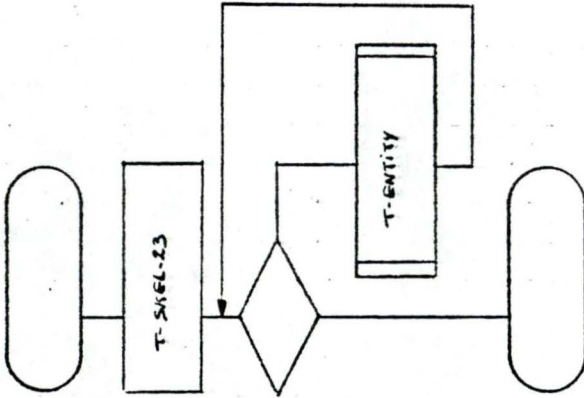
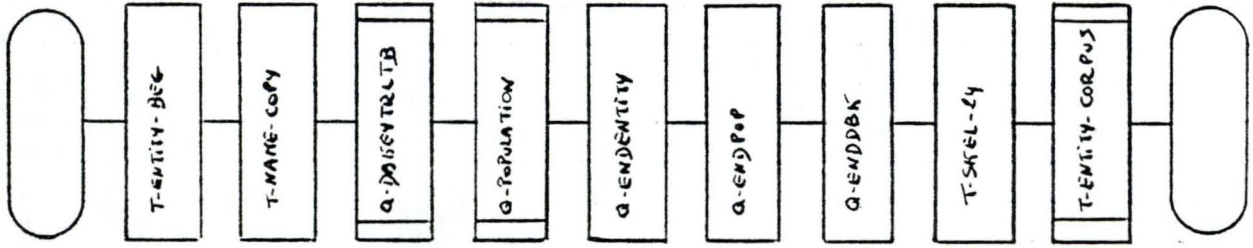
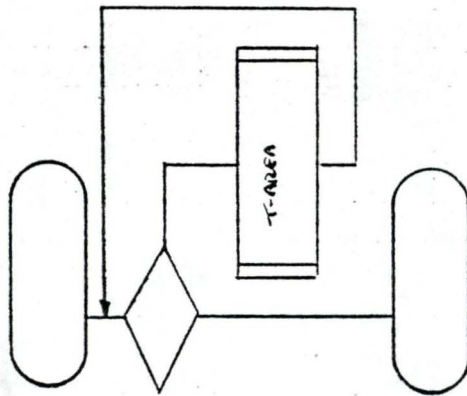
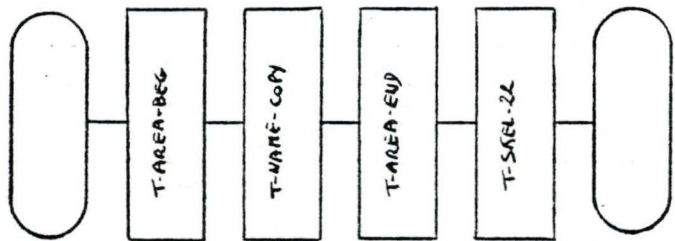
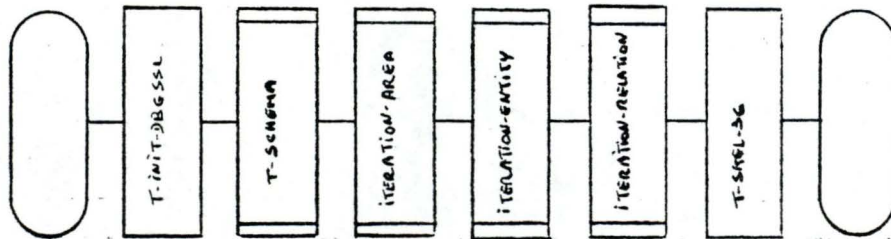
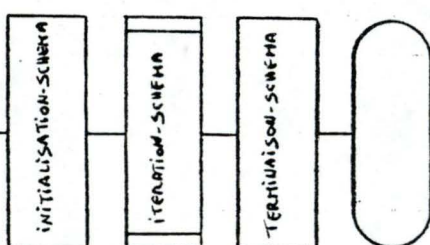
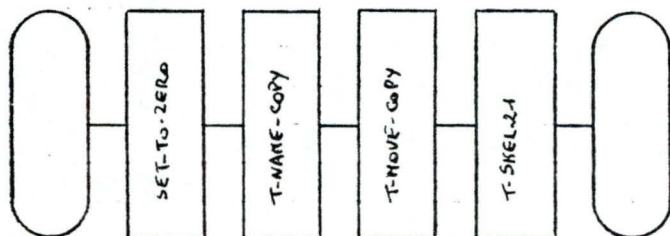
ROOT-OF-THE-PROGRAM INITIALISATION-SCHEMA	mise à zéro des diverses zones de travail, ouverture du fichier VIRMAN
ITERATION-SCHEMA TERMINAISON-SCHEMA T-INIT-DBGDDL	fermeture du fichier VIRMAN lecture de TBENTRYDIREC qui est la DIRECTORY pour cette génération clauses SCHEMA
T-SCHEMA ITERATION-AREA ITERATION-ENTITY ITERATION-RELATION SET-TO-ZERO	mise à zéro des zones de travail et des zones de valeurs pour PRGENSKE séparation du nom de l'objet et de son numéro de version
T-NAME-COPY	
T-MOVE-COPY	sauvetage des adresses début et fin des index d'entités et de relations
T-RESPRO-COPY	responsable problem-definer rangé dans la table des para- mètres pour PRGENSKE
T-RESINT-COPY	responsable interface rangé dans la table des paramètres pour PRGENSKE
Q-PRIVACY-COPY	clause PRIVACY ?
T-SKEL-1	
Q-OTHERPRIV-COPY	plusieurs littéraux ?
T-PRIVACY	clause PRIVACY
T-OTHERPRIV	autres littéraux dans clause PRIVACY ?
T-PRIV-BEG	
T-LITLTERAL	premier littéral dans clause PRIVACY
T-OTHER	autre littéral dans clause PRIVACY ?
T-PRIV-OTHER	} autre littéral dans clause PRIVACY
T-LITT-OTHER	
Q-OTHER-LITT-PRIV	
T-SKEL-2	
T-SKEL-3	
T-AREA	clauses AREA
T-SKEL-5	
T-AREA-BEG	
Q-TEMPORARY	area TEMPORARY ?
T-SKEL-4	
T-ENTITY	clauses ENTITY
FIRST-PART-ENTITY	
SECOND-PART-ENTITY	
T-ENTITY-DEB	
Q-LOCATION	clause LOCATION MODE ?
T-SKEL-6	
Q-OTHERLOC	plusieurs clés pour le CALC du LOCATION MODE ?
T-DUPL	"duplicates are allowed or not" pour LOCATION MODE ?
T-SKEL-8	

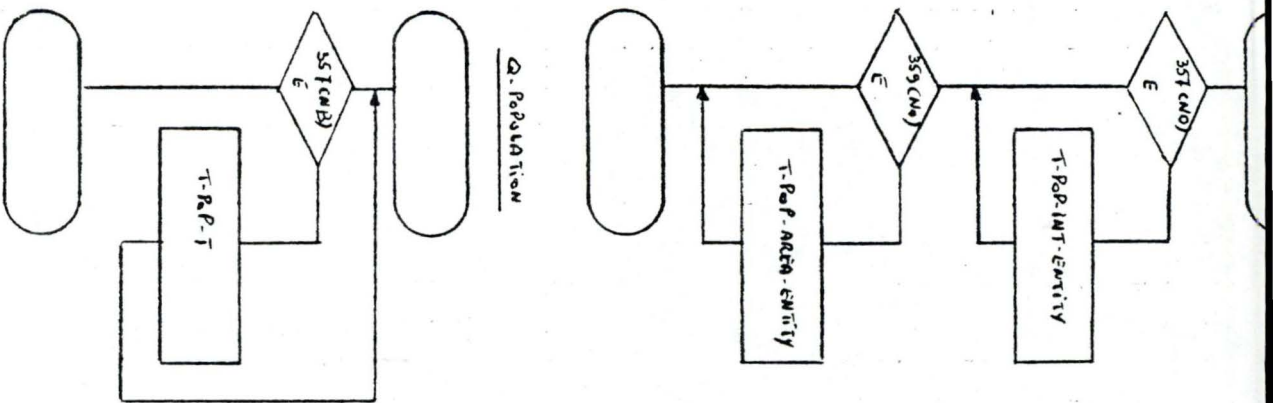
T-DESCRIPTION	description du contenu du RECORD
T-LOCATION	clause LOCATION MODE
T-DIREC	LOCATION MODE IS DIRECT
Q-CALC	LOCATION MODE IS CALC ?
T-CALC	LOCATION MODE IS CALC
T-ERR-LOC	erreur sur le type de LOCATION MODE
Q-PROC	procédure pour le CALC du LOCATION MODE ?
Q-VAL-CALC	plusieurs clés de CALC pour LOCATION MODE ?
T-PROC	procédure du CALC pour LOCATION MODE
T-VAL-CALC	plusieurs clés de CALC pour LOCATION MODE
T-ERR-VAL-LOC	erreur sur les valeurs du LOCATION MODE
T-OTHERLOC	
T-SKEL-7	
T-CALC-OTHER	différentes clés de CALC pour LOCATION MODE
T-WITHIN	clause WITHIN
Q-SEARCH	clause SEARCH ?
Q-GREL	présence de GROUP et/ou ELEMENT pour ce RECORD
Q-ENDENTITY	fin des clauses pour ce RECORD après clause WITHIN
T-WITHIN2	plusieurs noms d'AREA dans la clause WITHIN
T-ENDAREA	fin des noms d'AREA dans la clause WITHIN
T-SKEL-9	
T-SEARCH	clause SEARCH
T-SEARCH-SEC	
T-SEARCH-SEC-INIT	
T-SEARCH-SEC-CORPUS	
T-SEARCH-SEC-END	
Q-ENDSEARCH	fin des clauses pour ce RECORD après clause SEARCH ?
T-SKEL-11	
Q-DUPSEARCH	"duplicates are allowed" pour SEARCH ?
T-SKEL-12	
T-SEARCH-NAME	nom de la clé de SEARCH
T-S-S-C-2	
ERR-SEARCH	
T-SEARCH-IDX-NMD	nom de l'index dans la clause SEARCH
T-SKEL-13	
T-GREL	clauses de GROUP et/ou ELEMENT
T-NONIV	numéro de niveau
T-GREL-INIT	
Q-NONIV-COPY	numéro de niveau ?
Q-PICTURE-COPY	clause PICTURE ?
Q-TYPE	clause TYPE ?
Q-OCCURS-COPY	clause OCCURS ?
T-SKEL-14	
T-PICTURE	clause PICTURE
T-TYPE	clause TYPE
T-TYPE-INIT	
T-TYPE-INTEGGER-1	premier nombre entier dans la clause TYPE
T-TYPE-INTEGGER-2	second nombre entier dans la clause TYPE

T-OCCURS	clause OCCURS
T-SKEL-15	
T-RELATION	clause de RELATION
T-RELATION-INIT	
T-OWNER	owner de la RELATION
T-MEMBER-1	member de la RELATION (son nom)
T-ORDER	clause ORDER
T-SKEL-16	
T-MEMBER-2	clauses pour 1 member
Q-INDEXED	index dans la clause ORDER ?
Q-SORTED	"sorted" dans la clause ORDER ?
T-INDEXED	index dans la clause ORDER ?
T-SORTED	"sorted" dans la clause ORDER ?
T-SORTED-DUPL	"duplicates" dans la clause ORDER
T-NAMED	nom d'index dans la clause ORDER
T-MEMBER-2-INIT	
T-MEMBER-2-CORPUS	
T-MEMBER-2-END	
T-TYPAPPART	clause TYPE D'APPARTENANCE
T-SKEL-17	
Q-TYPKEY	clause KEY ?
T-TYPAP1	type d'appartenance (partie 1)
T-TYPAP2	type d'appartenance (partie 2)
Q-ENDREL	fin des clauses de cette RELATION ?
Q-ENDTYPAP	fin des clauses de cette RELATION après la clause TYPE D'AP-
T-TYPKEY	type de clé de la clause KEY
T-KEY-NAME	nom de la clé de la clause KEY
T-KEY	clause KEY
T-ENDKEY	fin des clauses de cette RELATION après la clause KEY ?
T-SKEL-18	
T-SEARCH-REL	clauses SEARCH pour la RELATION
T-SELECTION	clauses SELECTION
T-SEARCH-UNIT	
T-SEARCH-NAME	
T-SEARCH-USING	
T-SEARCH-DUPL	
T-S-NAME	nom de clé de la clause SEARCH
T-S-INIT	
T-SKEL-10	
T-END-SEARCH	fin des clauses de cette RELATION après la clause SEARCH ?
T-SKEL-12	
T-TYP-SEARCH	type de la clause SEARCH
T-IDX-SEARCH-NMD	nom d'index dans la clause SEARCH
T-PROC-SEARCH	nom de procédure dans la clause SEARCH
T-ERR-SEARCH	erreur dans la clause SEARCH (3e opt. CODASYL incorrecte en
T-SELECTION-INIT	UDS/V2)
T-SELECTION-END	
T-SKEL-19	
T-ALIAS	clause ALIAS dans la clause SELECTION
T-ALIAS-LEFT	
T-ALIAS-RIGHT	
T-ALIAS-END	
T-SKEL-20	

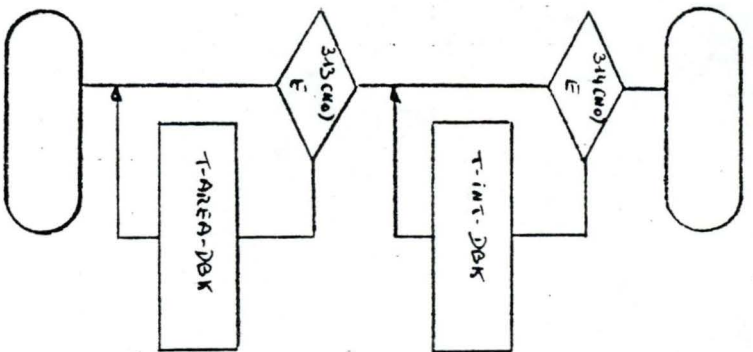


T-ENTITY

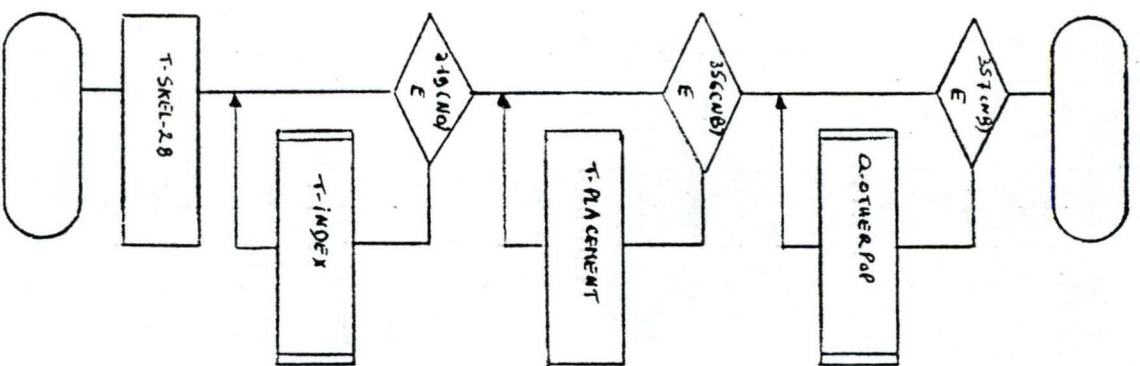
ITERATION-ENTITYITERATION-AREAT-AREAITERATION-SCHEMAROOT-OF-THE-PROGRAMT-SCHEMA



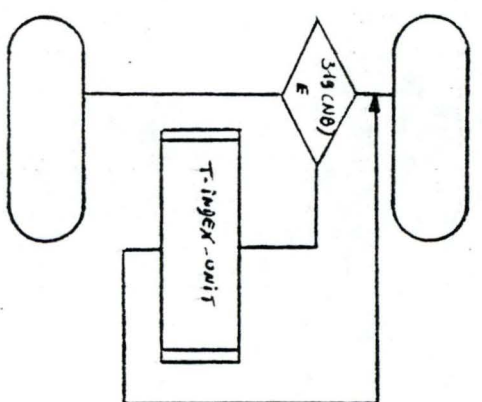
Q-POPULATION



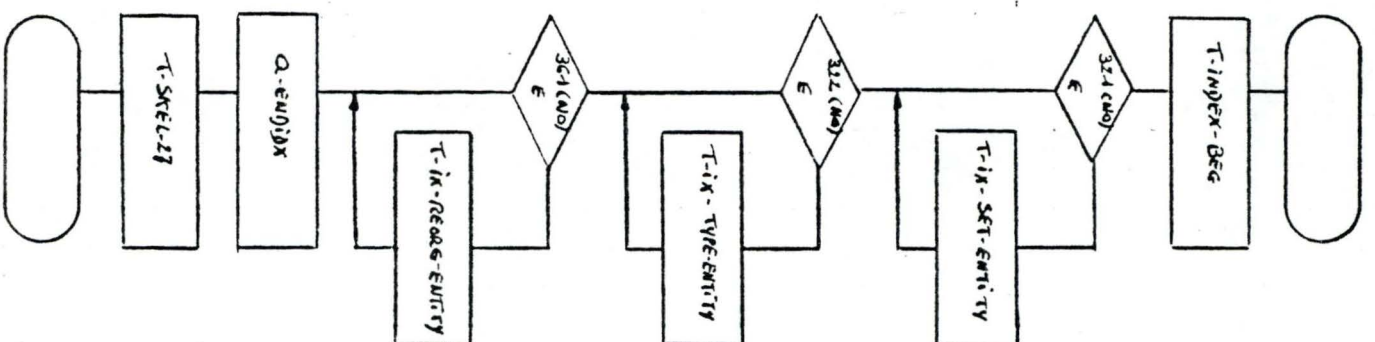
Q-DBKEYTRUTHB



T-ENTITY-CARDS



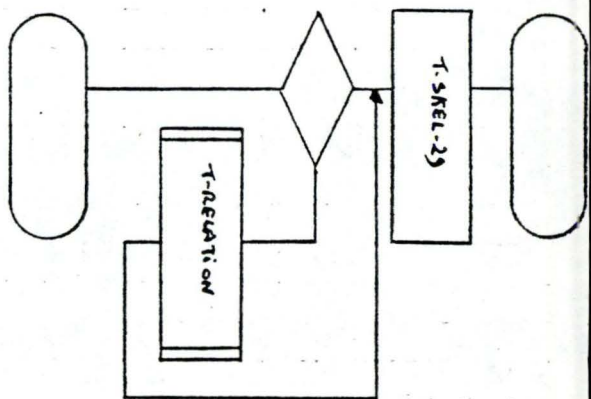
T-INDEX



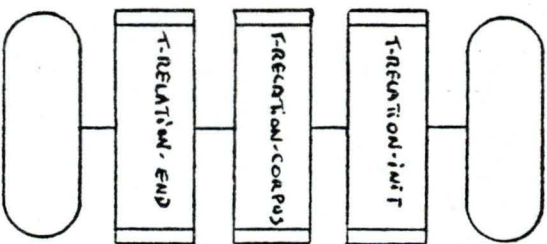
T-INDEX-UNIT

Q-OTHERPop

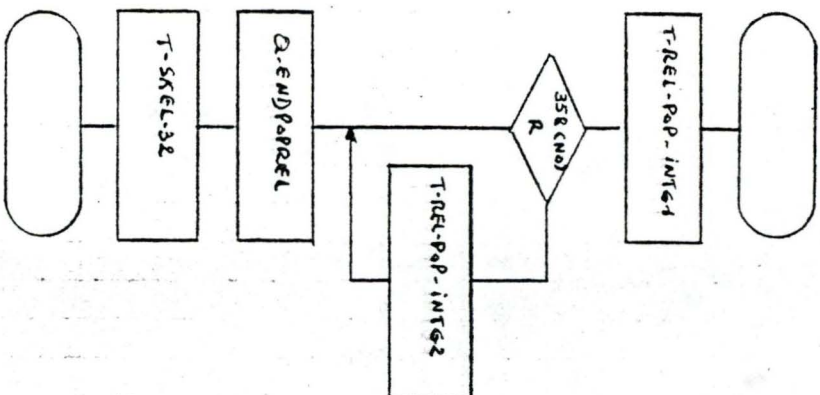
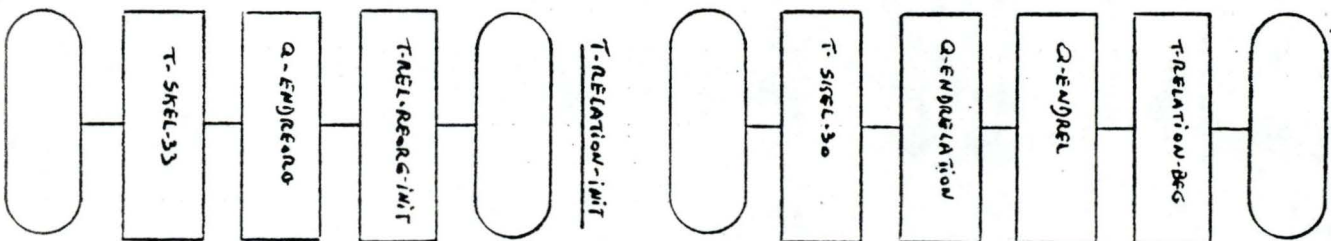




ITERATION-RELATION

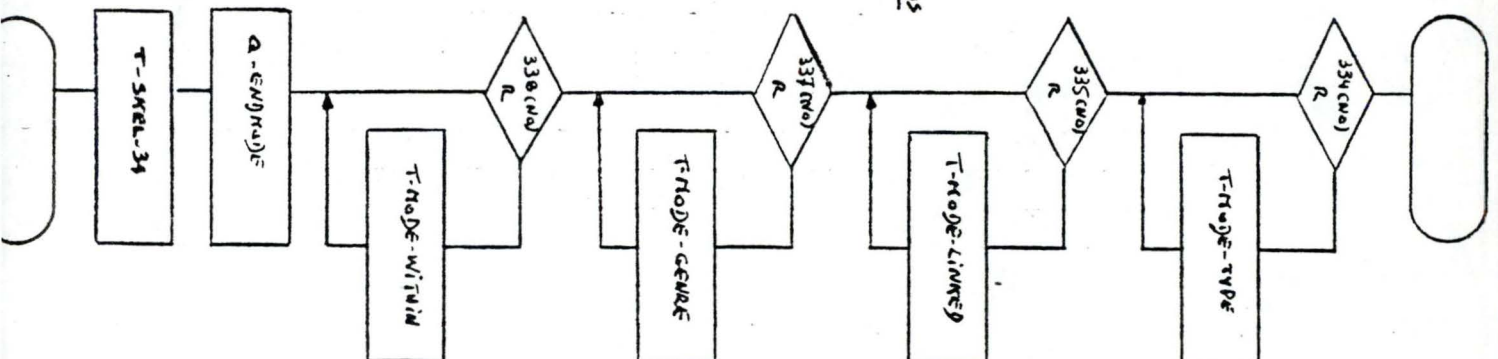


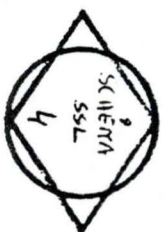
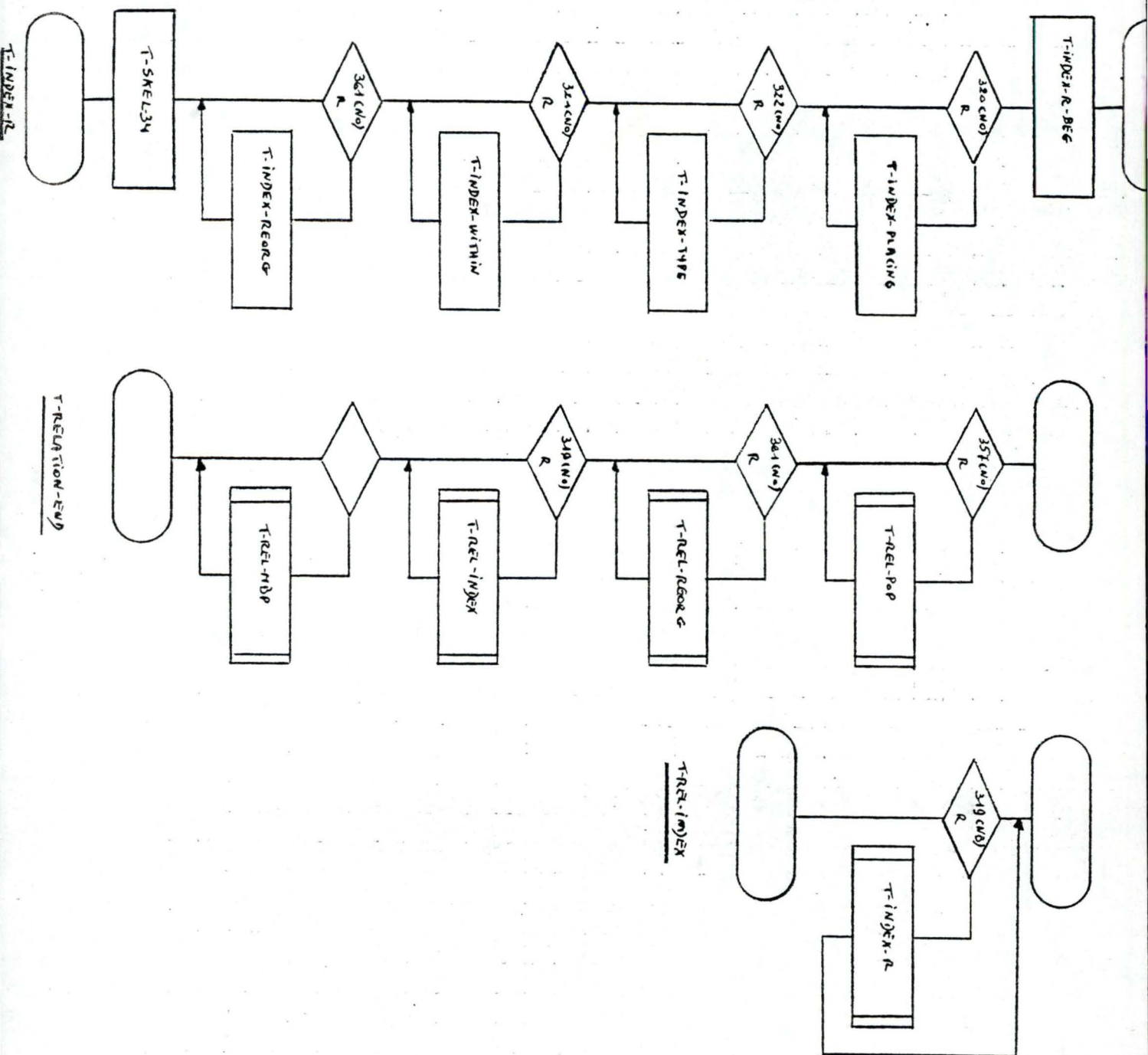
ITERATION-RELATION



T-REL-POP

T-RELATION-CORPUS

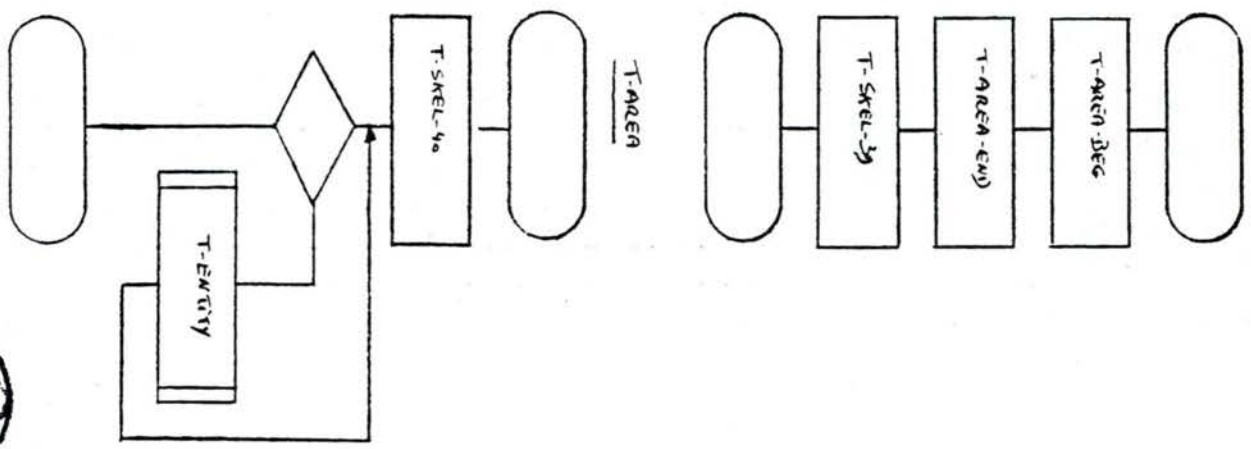
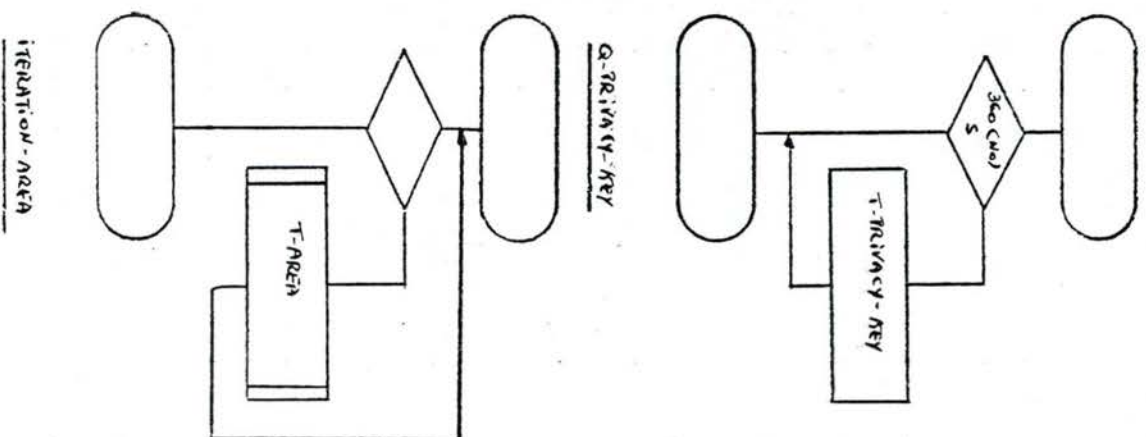
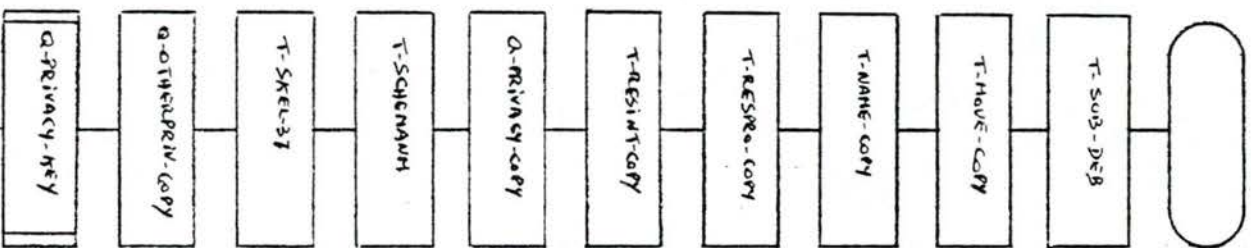
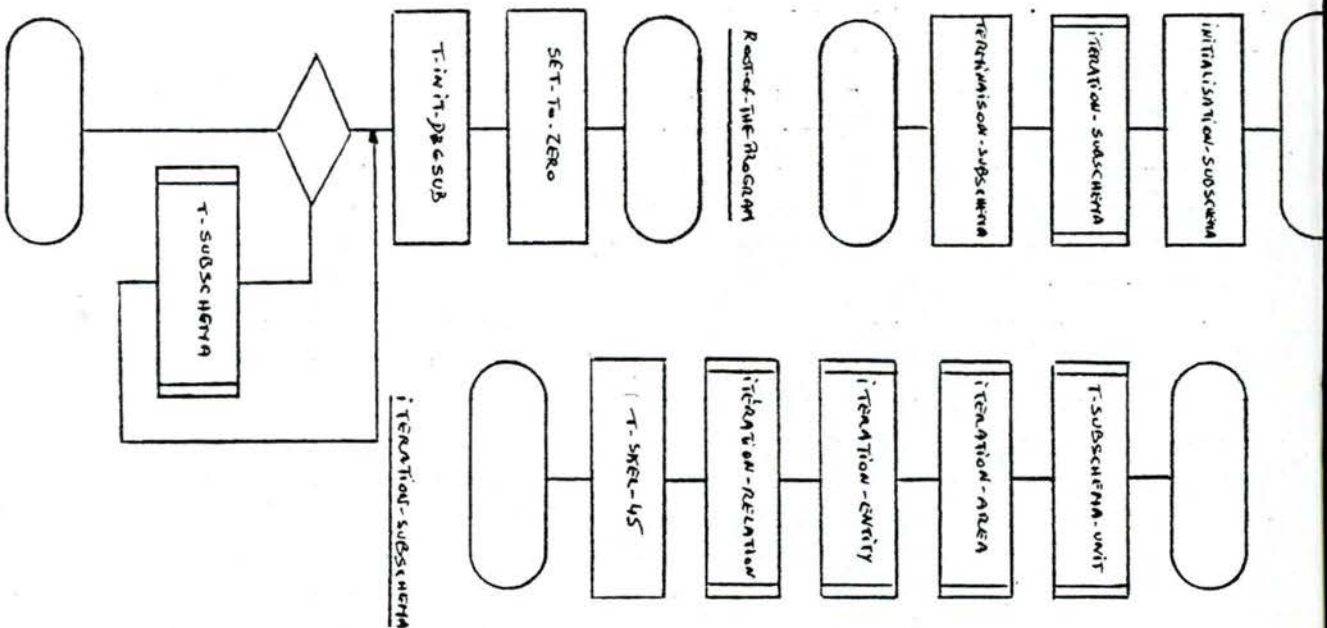


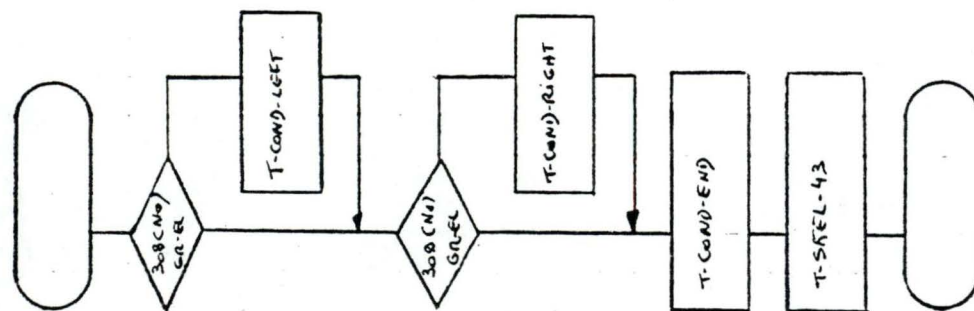


SCHEMA SSL (PRDBGSSL)

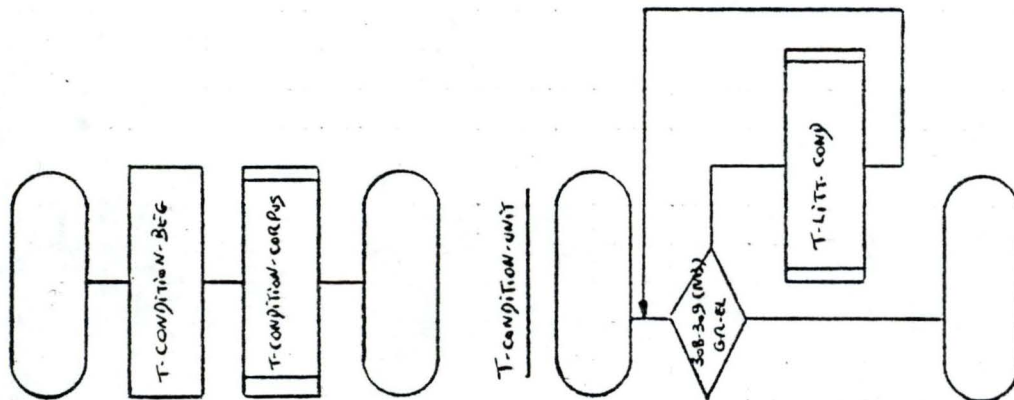
ROOT-OF-THE-PROGRAM	
INITIALISATION-SCHEMA	mise à zéro des diverses zones de travail, ouverture du fichier VIRMAN
ITERATION-SCHEMA	
TERMINAISON-SCHEMA	fermeture du fichier VIRMAN
SET-TO-ZERO	idem SCHEMA DDL
T-NAME-COPY	idem
T-SKEL-21	
T-INIT-DBGSSL	idem
T-SCHEMA	idem
ITERATION-AREA	
ITERATION-ENTITY	
ITERATION-RELATION	
T-SKEL-36	
T-AREA	clauses AREA
T-AREA-BEG	
T-NAME-COPY	idem
T-AREA-END	
T-SKEL-22	
T-SKEL-23	
T-ENTITY	clauses ENTITY
T-ENTITY-BEG	
T-NAME-COPY	idem
Q-DBKEYTRLTB	clause DATABASE-KEY-TRANSLATION-TABLE ?
Q-POPULATION	clause POPULATION
Q-ENDENTITY	fin des clauses pour ce RECORD ?
Q-ENDPOP	fin des clauses pour ce RECORD après clause DATABASE..?
Q-ENDDBK	fin clauses pour ce RECORD après clause POPULATION?
T-SKEL-24	
T-ENTITY-CORPUS	
T-POP-INT-ENTITY	entier de la clause POPULATION
T-POP-AREA-ENTITY	nom d'AREA de la clause POPULATION
T-POP-T	
T-INT-DBK	nombre entier de la clause DBKEY ... TABLE
T-AREA-DBK	nom d'AREA de la clause DBKEY ... TABLE
Q-OTHERPOP	autre clause POPULATION ?
T-PLACEMENT	clause PLACEMENT
T-INDEX	clause INDEX
T-SKEL-28	
T-INDEX-UNIT	
T-SKEL-25	

T-INDEX-BEG	
T-IX-SET-ENTITY	"placing" de la clause INDEX de l'ENTITY
T-IX-TYPE-ENTITY	"type" " "
T-IX-REORG-ENTITY	"reorganization" " "
Q-ENDIDX	fin des clauses du RECORD après la clause INDEX ?
T-SKEL-27	
T-SKEL-26	
T-SKEL-29	
T-RELATION	clauses de RELATION
T-RELATION-INIT	
T-RELATION-CORPUS	
T-RELATION-END	
T-RELATION-BEG	
Q-ENDREL	fin des clauses de cette RELATION ?
Q-ENDRELATION	fin des clauses de toutes les RELATIONS ?
T-SKEL-30	
T-MODE-TYPE	type de MODE
T-MODE-LINKED	"linked" dans la clause MODE
T-MODE-GENRE	genre du MODE
T-MODE-WITHIN	"within" dans la clause MODE
Q-ENDMODE	fin des clauses de RELATION après la clause MODE ?
T-SKEL-31	
T-REL-POP	clause POPULATION
T-REL-REORG	clause REORGANIZATION
T-REL-INDEX	clause INDEX
T-REL-MBP	clause MEMBER IS PHYSICALLY ...
T-REL-REORG-INIT	
Q-ENDREORG	fin des clauses de RELATION après la clause REORGANIZATION?
T-SKEL-33	
T-REL-POP-INTG1	premier entier de la clause POPULATION
T-REL-POP-INTG2	second entier de la clause POPULATION
Q-ENDPOPREL	fin des clauses de RELATION après la clause POPULATION ?
T-SKEL-32	
T-INDEX-A	index de RELATION
T-SKEL-35	
T-INDEX-R-BEG	
T-INDEX-PLACING	"placing" dans la clause INDEX de RELATION
T-INDEX-TYPE	"type" dans la clause INDEX de RELATION
T-INDEX-WITHIN	"within" dans la clause INDEX de RELATION
T-INDEX-REORG	"reorganization" dans la clause INDEX de RELATION
T-SKEL-34	



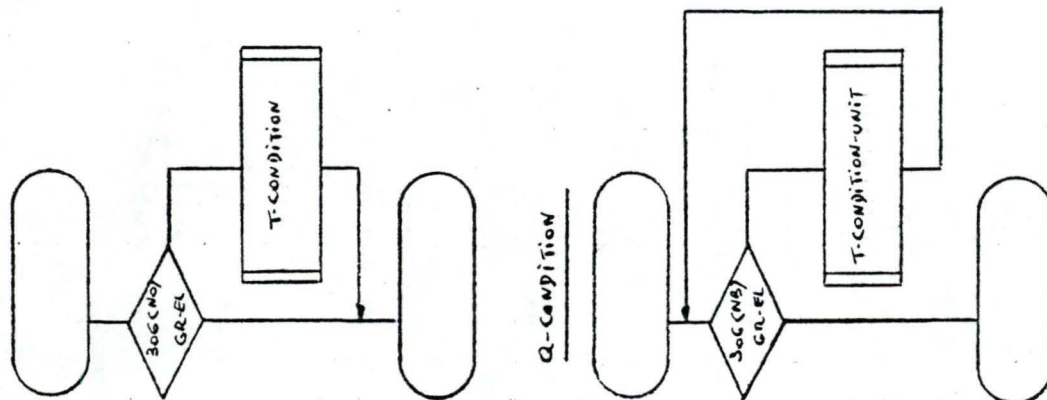


T-LITT-COND



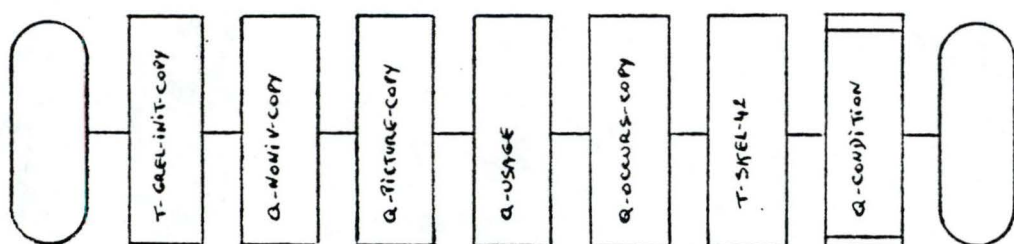
T-CONDITION-UNIT

T-CONDITION-CORPUS

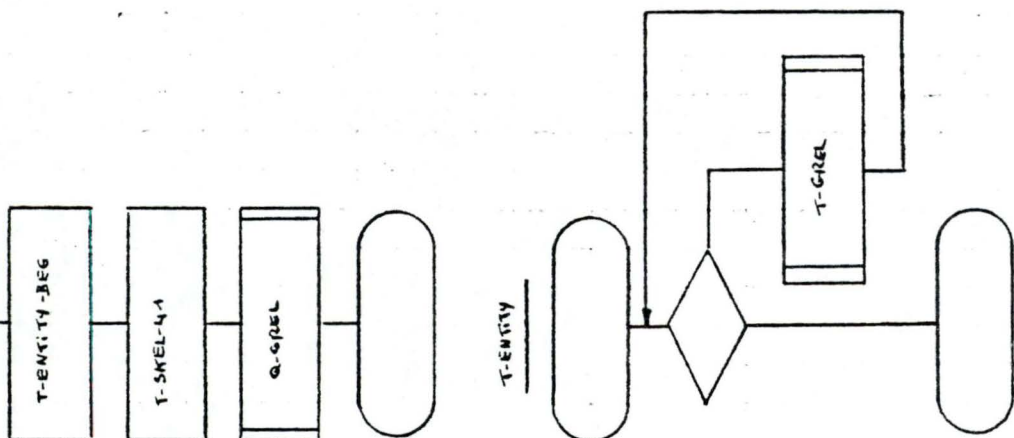


Q-CONDITION

T-CONDITION

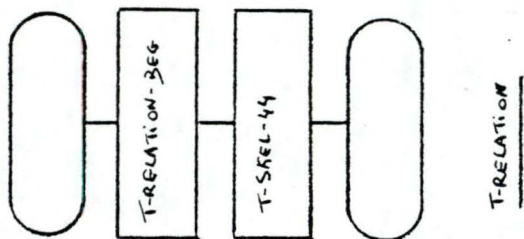
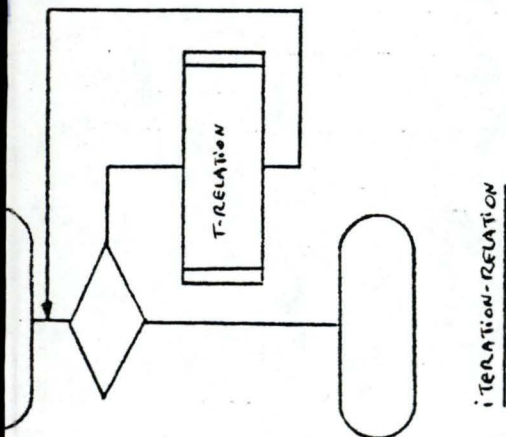


T-GREL



T-ENTITY

Q-GREL



SUB - SCHEMA (PRDBGSUB)

ROOT-OF-THE-PROGRAM
 INITIALISATION-SUBSCHEMA
 ITERATION-SUBSCHEMA
 TERMINAISON-SUBSCHEMA
 SET-TO-ZERO
 T-INIT-DBGSUB
 T-SUBSCHEMA
 T-SUBSHEMA-UNIT
 ITERATION-AREA
 ITERATION-ENTITY
 ITERATION-RELATION
 T-SKEL-45
 T-SUB-DEB
 T-MOVE-COPY
 T-NAME-COPY
 T-RESPRO-COPY
 T-RESINT-COPY
 Q-PRIVACY-COPY
 T-SCHEMANM
 T-SKEL-37
 Q-OTHERPRIV-COPY
 Q-PRIVACY-KEY
 T-PRIVACY-KEY
 T-AREA
 T-AREA-BEG
 T-AREA-END
 T-SKEL-39
 T-SKEL-40
 T-ENTITY
 T-ENTITY-BEG
 T-SKEL-41
 Q-GREL
 T-GREL
 T-GREL-INIT-COPY
 Q-NONIV-COPY
 Q-PICTURE-COPY
 Q-USAGE
 Q-OCCURS-COPY
 T-SKEL-42
 Q-CONDITION
 T-CONDITION
 T-CONDITION-UNIT
 T-CONDITION-BEG
 T-CONDITION-CORPUS
 T-LITT-COND
 T-COND-LEFT
 T-COND-RIGHT
 T-COND-END
 T-SKEL-43
 T-RELATION
 T-RELATION-BEG
 T-SKEL-44

mise à zéro des zones de travail, ouverture du fi-
 chier VIRMAN
 fermeture du fichier VIRMAN
 idem SCHEMA DDL
 idem
 idem
 idem
 idem
 idem
 idem
 idem
 idem
 clause SUB-SCHEMA
 idem
 clause PRIVACY KEY ?
 clause PRIVACY KEY
 clause AREA
 clauses ENTITY
 clauses pour GROUP et/ou ELEMENT ?
 clauses pour GROUP et/ou ELEMENT
 idem
 idem
 idem
 clause USAGE ?
 idem
 clause CONDITION ?
 nom de CONDITION
 littéral de valeur de CONDITION
 littéral de valeur de CONDITION (partie début fourche)
 littéral de valeur de CONDITION (partie fin fourche)
 fin des clauses de CONDITION
 clause RELATION

SCHEMA DDL	UDS/V2
------------	--------

SCHEMA NAME IS schema-name

[PRIVACY LOCK FOR COPY IS litteral [OR litteral [OR litteral]]].

AREA NAME IS area-name

[AREA IS TEMPORARY].

RECORD NAME IS record-name

[LOCATION MODE IS { DIRECT { d-b-d-n
d-b-i
CALC [d-b-procédure] USING d-b-d-n [, d-b-d-n] ...
DUPLICATES ARE [NOT] ALLOWED } }]

WITHIN area-name [{ area-name } ... AREA-ID IS d-b-i]

[SEARCH KEY IS d-b-d-n [, d-b-d-n] ... USING { CALC [d-b-procédure]
INDEX [NAME IS index-name] }
DUPLICATES ARE [NOT] ALLOWED] ...]

[n° niveau] d-b-d-n

[{ PIC
PICTURE } IS { d-b-d-n
d-b-i }]

[TYPE IS { || BINARY
DECIMAL
FIXED
REAL
CHARACTER
DATABASE-KEY || integer [, integer] }]

[OCCURS integer TIMES] .

SET NAME IS set-name

[SET IS DYNAMIC]

ORDER IS

FIRST

LAST

NEXT

PRIOR

IMMATERIAL

SORTED [INDEXED [NAME IS index-name]]

{ BY DATABASE-KEY

BY DEFINED KEYS

DUPLICATES ARE [NOT] ALLOWED }

OWNER IS { record-name
SYSTEM }

MEMBER IS record-name { MANDATORY
OPTIONAL } { AUTOMATIC
MANUAL }

[{ ASCENDING
DESCENDING } KEY IS d-b-d-n [, d-b-d-n] ...]

[SEARCH KEY IS d-b-d-n [, d-b-d-n] ... USING { CALC [d-b-procédure]
INDEX [NAME IS index-name] }
DUPLICATES ARE [NOT] ALLOWED ...]

[SET OCCURENCE SELECTION IS THRU { CURRENT OF SET
LOCATION MODE OF OWNER
[ALIAS FOR { d-b-d-n
d-b-i } IS d-b-d-n] ... }] .

STORAGE STRUCTURE OF SCHEMA schema-name.

AREA NAME IS area-name IS ASSIGNED TO DCB-NAME DCB-name.

RECORD NAME IS record-name

[DATABASE-KEY-TRANSLATION-TABLE [IS integer] [WITHIN area-name]
[POPULATION IS integer WITHIN area-name [, integer WITHIN area-name] ...

[PLACEMENT OPTIMIZATION FOR SET set-name]

[INDEX NAME IS index-name [PLACING IS WITHIN area-name]
[TYPE IS { DATABASE-KEY-LIST
REPEATED-KEY }]

[DYNAMIC REORGANIZATION SPANS integer PAGES]] ...

[COMPRESSION FOR ALL ITEMS] .

SET NAME IS set-name

[MODE IS { CHAIN [LINKED TO PRIOR]
{ POINTER-ARRAY } { ATTACHED TO OWNER
LIST { DETACHED WITHIN area-name] } }
[WITH PHYSICAL LINK] }]

[POPULATION IS integer [INCREASE IS integer]]

[DYNAMIC REORGANIZATION SPANS integer PAGES]

[INDEX NAME IS index-name [PLACING IS { ATTACHED TO OWNER
DETACHED [WITHIN area-name] }]]

[TYPE IS { DATABASE-KEY-LIST
REPEATED-KEY }]

[DYNAMIC REORGANIZATION SPANS integer PAGES]] ...

[MEMBER IS PHYSICALLY LINKED TO OWNER] .

SUB-SCHEMA DDL UDS/V2

SUB-SCHEMA NAME IS subschema-name OF SCHEMA NAME schéma-name

[PRIVACY LOCK FOR COMPILE IS literal [OR literal [OR literal]]]

[PRIVACY KEY FOR COPY IS 'literal'] .

AREA SECTION.

COPY { ALL AREAS
area-name [, area-name] ... } .

RECORD SECTION.

01 record-name

n° niveau d-b-d-n

[PICTURE IS d-b-i]

[USAGE IS { COMPUTATIONAL
COMPUTATIONAL-3
DISPLAY
DATABASE-KEY }]

[OCCURS integer TIMES] .

88 condition-name { VALUE IS } littéral [{ THRU
THROUGH } littéral]

[VALUES ARE]
[littéral [{ THRU
THROUGH } littéral] --- .

SET SECTION.

COPY { ALL SETS
set-name [, set-name] ... } .

TABLE DES PARAMETRES

NOM	N° ENTREE	LG. REELLE	SIGNIFICATION
NAME	1	12	nom de l'objet
RESPRO	2	12	nom du responsable-problem-definer
RESINT	3	12	nom du responsable-interface
VERS	4	2	n° de version
LITT	5	1-20	littéral
FRDES1	6	1-20	description en français (partie 1)
FRDES2	7	0-20	description en français (partie 2)
FRDES3	8	0-20	description en français (partie 3)
FRDES4	9	0-20	description en français (partie 4)
NTDES1	10	1-20	description en neerlandais (partie 1)
NTDES2	11	0-20	description en neerlandais (partie 2)
NTDES3	12	0-20	description en neerlandais (partie 3)
NTDES4	13	0-20	description en neerlandais (partie 4)
KEYLOC	14	12	nom de clé du LOCATION MODE CALC ou DIRECT
PROCNM	15	12	nom de procédure
INDXNM	16	12	nom d'index
NONIV	17	2	n° de niveau
INFPIC	18	1-20	informations de PICTURE
TYPCAR	19	1 ou 10	type de l'élément
ENTIER	20	1- 6	nombre en "integer"
FACTRP	21	1- 6	facteur de répétition
OWNER	22	12	nom de l'owner
MEMBER	23	12	nom du member
TYPSOR	24	1 ou 16	type de "sorted" dans ORDER
TYPKEY	25	1 ou 10	type de clé (ASC/DESCENDING)
TYPAP1	26	1 ou 10	type d'appartenance (partie 1)
TYPAP2	27	1 ou 10	type d'appartenance (partie 2)
NEWNM	28	12	nouveau nom pour 1 objet
DCBNM	29	12	nom du DCB de l'area
AREANM	30	12	nom d'un area
REALNM	31	12	nom d'un realm
SCHENM	32	12	nom d'un schéma
INFUSG	33	1 ou 10	informations de USAGE
ENTR2	34	1- 6	nombre en "integer"
MODE	35	1 ou 16	type de mode
LITT2	36	1-20	littéral
ERR1	37	1-20	libellé d'erreur (partie 1)
ERR2	38	1-20	libellé d'erreur (partie 2)
ORDER	39	1 ou 20	type d"order"

TABLE DES CONDITIONS

NOM	N° ENTREE	VALEUR	SIGNIFICATION
END1	1	1	switch de fin (n°1)
END2	2	1	switch de fin (n°2)
IMP	3	1	switch d'impression
TYP1	4		switch sur le type de ...
PRIVAL	5	1	clause PRIVACY
TEMPOR	6	1	clause TEMPORARY
PROCED	7	1	présence d'une procédure
DUPLNO	8	1	DUPLICATES ARE <u>NOT</u> ALLOWED
NIVEAU	9	1	présence d'1 n° de niveau
DBKEYT	10	1	clause DATABASE-KEY-TRANS-TABLE
LINKED	11	1	linked to owner
USAGE	12	1	clause USAGE
OTHERP	13	1	présence d'autres littéraux de PRIVACY
LOCCAL	14	1	location mode IS CALC
INDEX	15	1	présence d'1 "index-name"
INTEGR	16	1	présence d'1 "integer"
REORG	17	1	clause réorganisation
MODE1	18	1	mode de type ATTACHED
LOCDIR	19	1	location mode IS DIRECT
PICTUR	20	1	clause PICTURE
DYNAM	21	1	clause DYNAMIC REORGANIZATION
REAL	22	1	présence d'1 realm-name
TYPIDX	23		type d'index
MODE2	24	1	mode de type DETACHED
PLUSLT	25	1	plusieurs littéraux
OCCRS	26	1	clause OCCURS
NAMED	27	1	index nommé
POPUL	28	1	clause POPULATION
PLACEM	29	1	clause PLACEMENT
MODE3	30	1	mode WITH PHYSICAL LINK
OTHERL	31	1	présence d'autres littéraux
TYPE	32	1	clause TYPE.
DIFFDUP	33		switch sur le duplicateur
WITHIN	34	1	clause WITHIN
DUPL	35	1	clause DUPLICATES
PLACM2	36	1	placement (2e partie)
SORTED	37	1	clause ORDER IS SORTED

TABLE DES POSITIONS

<u>NOM</u>	<u>N° ENTREE</u>	<u>SIGNIFICATION</u>
POSIT1	1	valeur de position (n°1)
POSIT2	2	valeur de position (n°2)
POSIT3	3	valeur de position (n°3)
POSIT4	4	valeur de position (n°4)
POSIT5	5	valeur de position (n°5)

BUMP



0 0 3 2 1 2 9 3 6

*FM B16/1979/17

